Species Classification on Thermal Video Using a Convolutional Recurrent Neural Network

Things That Go Bump In The Night

.....

A thesis Submitted in fulfilment of the requirements for the degree of

Master of Science

at

University of Canterbury

by

Christopher David Carr

Supervised By: Prof. Richard Green

August 2021

Abstract

This paper proposes a new approach to species surveying, utilising convolutional recurrent neural networks (CRNNs). By using breakthroughs in neural network architectures and designs, as well as modern hardware, new approaches are possible that have not yet been investigated. Analysing thousands of hours of footage allows for more accurate, timely, and interesting surveying footage, far surpassing current approaches used by conservation programs. Prior to this research, a reliable dataset of thermal images did not exist, much less a dataset that records motion. Further, the data has been labelled, and categorised by location and time. While the creation of this dataset alone is a contribution, the CRNN has a high performance and reliable detection for all trained classes, which increases as more data is gathered. This puts this neural network approach ahead of any other extant method, as those that do exist either use static images, infrared illumination, or perform worse.

The proposed approach is much better at detecting animals than current low tech trap or observation based approaches (by over 3 thousand times), such as trapping lines, transects, dog hunting, or observations. Further, it is more accurate than extant trail cameras for detecting small mammals - being about 10-50 times better in experimental trials.

Furthermore the net itself performs well on trained classes, with the accuracy of the CRNN reaching up to 87 percent and the catchment includes all night hours (the definition of which can be increased or decreased based on latitude and time of year, or simply ambient light levels) - and the filming technique uses a thermographic passive infrared camera, and requires a cold background. Processing time (per occurrence) is unaffected by total footage (3ms processing time per animal-occurrence), though obviously the more footage captured, the more that needs to be processed, also increasing linearly. Finally, the approach described in this paper has the potential to be used internationally, on all continents and environments, limited only by the annotated dataset size and quality on which it is trained, on all animals over a certain size, whether those animals interact, are delicate/easily damaged, or rare. While not being

proposed as a replacement for all of the existing manual quantification tools, it that been shown to be a successful and useful addition added to the toolkits of conservation efforts.

Table Of Contents

List of Figures

List of Tables

1 Introduction

"This morn I was awakd by the singing of the birds ashore from whence we are distant not a quarter of a mile, the numbers of them were certainly very great who seemd to strain their throats with emulation perhaps; their voices were certainly the most melodious wild musick I have ever heard, almost imitating small bells but with the most tuneable silver sound imaginable."

Botanist Joseph Banks, 1770, On board the Endeavor [6]

Captain James Cook also described the dawn chorus of New Zealand (heard from his ship, anchored offshore) as "positively deafening"[6]. New Zealand had existed in a unique state before people arrived; having no warm blooded mammalian predators. This state had been the status quo since New Zealand separated from the Australian continent 85 million years ago, and due to this, many of the animals on the island became specialized foragers with only a few true native predators, all of which were also birds. The introduction of a number of new species, both accidental and intentional by settlers from the Polynesian islands and Europe created a large number of problems for the vulnerable national bird population, who were poorly adapted for survival from these new threats as shown in Figure 1. The Department of Conservation in New Zealand is single-minded in its intent towards introduced predator species - to create a "Predator Free New Zealand"[19]. As such, various methods are being used to control populations of pest species, such as trapping, pest-proof fences, and 1080 poison drops [50].

Kate Gudsell, New Zealand Radio, New Zealand

Figure 1: New Zealand Bird Species

The long term goal of this project is to develop a tool which can be placed in New Zealand bush, which will enhance the efforts of the Department of Conservation's attempts towards controlling invasive pest species such as possums, mustelids (weasels, stoats, ferrets), pigs, rodents, hedgehogs and feral cats. These species can have a drastic and detrimental effect on native flora and fauna, and as such their control and eventual eradication is a high priority in New Zealand. The scope of this report is primarily constrained to the implementation of an AI - in this case a neural network - for the purposes of identification of animals.

Kate Gudsell, New Zealand Radio, New Zealand

Figure 2: New Zealand (Schedule 5) Pest Species

2 Background

2.1 Invasive Species Definition

In the USA exective order 13112 of February 3, 1999 (Invasive Species) an invasive species is defined as "an alien species whose introduction does or is likely to cause economic or environmental harm or harm to human health". Whether an animal becomes classified as invasive depends on innumerable factors, from its traits, The inter and intra-specific interactions, the vulnerability and fragility of the ecosystem and its inhabitants, specific timing of plenty or drought, human movements and industry. Species that do become invasive can have a profound and destructive effect on industry, agriculture, environmental and conservation efforts, and human health. New Zealand has its own Wildlife Act[73], and describes a number of animals explicitly, However nowhere is declared which species are "pests", which are "invasive", which are "naturalised", and which are "non-native". Those not explicitly

declared are protected by default. There are two main schedules that are primarily relevant:

Schedule 5 - Wildlife not protected. This is a large group that includes many common domestic and introduced species, many of which are regarded as pests. It includes numerous land mammals and birds, three species of Australian Litoria tree frogs, the Australian rainbow skink (Lampropholis delicata) and the North American red-eared slider turtle (Trachemys scripta elegans). The only species on this list that occur naturally in New Zealand are the southern black-backed gull and the spur-winged plover (masked lapwing), both of which present a significant risk of bird strike.

Schedule 6: Animals declared to be noxious animals subject to the Noxious Animals Act 1956. This group consists of the chamois, the Himalayan tahr, and all species of deer (the family Cervidae), goats (the genus Capra), and pigs (the genus Sus). All are considered harmful to New Zealand's native forests and may be hunted without restriction[73].

The critical part here is that all animals on schedules 5 and 6 are considered "not protected" - and although not all equally destructive, they are all considered unwelcome in New Zealand forests. Predator Free 2050[19] has defined success as eradication as elimination of Stoats (a classification which includes all mustelids), rats, and possums - all of which fall under Schedule 5 as shown in Figure 2.

2.2 Invasive Species' Threat to New Zealand

Environmental concerns in New Zealand are the primary motivation and focus of this research; establishing new tools and comparing with existing methods is one of the best ways to increase our environmental conservation efforts. New Zealand is incredibly vulnerable to disruption from invasive species for a number of reasons. Paini et al. perform a purely economic analysis[46] analysis of loss due to invasive species, and although this is a marginally old paper (2016) it is a robust comparison of the prospective economic costs. New Zealand alone stands to lose up to $639.7 million dollars from its agricultural exports, or .66% of GDP

- this makes invasion costs (economically, to agriculture alone) worse for NZ than Canada (at .63%), France (at .61%), Japan (at .52%), the UK (at .19%), Ireland (at .15%), and numerous other countries[46]. The threat, of course, is not purely economic, but also that of the loss of species due to invasion by other (pest) species. Currently, well over 75% of terrestrial animals in New Zealand are endangered[62].

- 84 percent of reptile species (89 of 106)
- 80 percent of bat species (4 of 5)
- 75 percent of frog species (3 of 4)
- 74 percent of terrestrial bird species (78 of 105)

These numbers are also not expected to improve over time:

- 46 percent of reptile species populations are expected to decrease (49 of 106), compared with 5 percent expected to increase (5 of 106)
- 60 percent of bat species populations are expected to decrease (3 of 5), compared with 20 percent expected to increase (1 of 5)
- 50 percent of frog species populations are expected to decrease (2 of 4), with none expected to increase (0 of 4)

It is impossible to definitively say what the cause of this incredible vulnerability of NZ species is, and it is beyond the scope of this project to fully investigate all of the economic, cultural, and environmental damages that every one of the pest species may inflict on New Zealand. However, it is clear that NZ animals are vulnerable to extinction, and that there is very real damage being inflicted by pest species in New Zealand.

2.3 Monitoring Species Abundance

Observing animals in the wild is a key part of ecology. Species abundance is the number of individuals belonging to a species in a region - this value is usually derived from a ratio based on sampling limitations such as the number of bellbirds heard in 10 minutes. Species abundance is used for conservation, ecology, surveying, species abundance deltas, and anthropogenic impact. These observations are especially useful when they are able to be reliably compared with respect to time; as one of the primary goals of human conservation efforts is rebuilding endangered species numbers. An increasingly large area of the planet is being affected by human interaction, impacting behaviour and habitat of animals.

There are a number of different monitoring tools currently being used by the Department of Conservation (DOC). Many of these traps require interaction from the target species, such as requiring that animal to directly engage via chewing, scratching, or transversing. Others require human monitoring, which comes with a whole host of problems in and of itself, such as reliability, replicability, specific timing, and intense man-hours to generate useful results. None of these tools are ideal, as they are sensitive to the behaviour of the animals themselves, as well as the observer.

Approach	**Target Species**
Night Counts	Rabbits
Trap Catch	Possums
Snap Traps	Rodents
Tracking Tunnels	Small Mammals (rodents and Mustelids)
Faecal Pellet Counts	Deer
Wallaby Counts	Bennett's Wallaby
True Census	Forest Birds in restricted area
Ground Photo Counts	Seabirds (colonial ground nesting birds)
Aerial Photo Counts	Seabirds (especially for remote nesting, such as offshore islands)
Five Minute Bird Count	Forest Bird Relative Abundance
Line Transect Count	Forest Bird Relative Abundance
Mist Netting	Density and Demography of bird populations

Table 1: Current Methods of Species Monitoring[17][18]

The tools that require direct interaction are unreliable for the simple fact that each animal behaves differently as shown in Table 1. Individuals do not always interact with the tools. Either out of fear, confusion, misunderstanding (not chewing on wax tags themselves), lack of motivation (going around the trap rather than over it), or simply not bumping into the trap (some traps are unbaited, and even those that are baited have a relatively limited effective "attractive" range, limited by the olfactory sense of the pest in question)[10, 14, 67].

The change in prey species' populations is used as a proxy for pest species populations, however monitoring prey species has shortcomings of its own; an accurate measure of prey species is difficult for the same reasons as monitoring pest species, and the changes in prey species can be caused by other factors such as different predators (that were not the ones being "controlled"), human activity, environmental, or pathogenic causes. However, mon-

itoring prey species does have its place, particularly in New Zealand, where our primary concern is increasing the number of our native bird and insect species, as opposed to strictly reducing the number of pest species (which is to say, if New Zealand somehow increased the number of native birds, without any noticeable change in the number of pests, that would be a net-positive).

Human gathered data (i.e. a person with a recording device) is not a good way to collect large volumes of accurate, standardised data. Data collected from humans will inherently be limited by the number of people in the area (which should be minimised to prevent damage), the quality of the photography (whereby not all footage will capture the relevant details), accuracy (whereby not all data will be correctly catalogued, labelled, or accessible), and standardisation (whereby the footage will be from different positions, angles, devices, and times). However, human gathered data may be used to refute the null hypothesis of animal presence, in that only one reliable sighting is needed to assert that an animal is present.

2.4 Human Impacts on Ecosystems

Ecological and Environmental groups across the planet have set the monumental task of arresting the spread and damage caused by invasive and introduced species[20]. As part of its *Threatened Species Strategy*, Australia plans to kill two million feral cats *Felis catus* and aims to control invasive common carp *Cyprinus carpio* by releasing a virus across one million square kilometres.[42]. The United States Fish and Wildlife Service has been proposing unlimited collection of invasive species of fish. Further, and of the most significant note locally, New Zealand has pledged funding to reach predator free status by 2050[52].

The fast growth of humans and over-exploitation of natural resources causes rapid, substantial, and novel changes to the Earth's ecosystems, which inevitably have an often negative impact on species population. Many species have been driven to extinction, and many species have been inadvertently introduced to environments where they can massively damage delicate ecosystems [65]. The latter case is of particular interest in New Zealand, as New Zealand

has a very isolated ecosystem (as an island nation), as well as a historically delicate one, with small introductions of new animals leading to explosive population growth. Further, damage introduced pests and predators have done is well known. It is therefore important for conservationists and researchers to accurately judge the number of animals in an area and the change in species populations over time in order to make informed judgements about conservation and management strategies.

2.5 Genetic Biocontrol agents

The release of organisms with genetic methods to disrupt reproduction is called Genetic Biocontrol, and is currently one of the prevailing methods for hugely integrated and populous dangerous species, such as mosquitoes, rats, pigeons. In many of these situations widespread poisoning of trapping is impossible without large collateral or huge costs.Emerging biotechnological control agents (notably CRISPR/Cas9 gene drives[44]) may increase the feasibility of widespread eradication on a land-mass (both island and continental) scale. This technology allows new approaches to increase genes with negative fitness to the population of a targeted species.

There are a number of technologies currently being used for species control at the genetic level. The four most prevalent techniques are sterile insect technique[33], YY Males[27], Trojan Female Technique[24], and gene drive[60]. Although outside the scope of this paper, cursory understanding of these techniques is important as they are all useful tools in species control, and any solutions proposed here should be considered within the context of these solutions as well.

- **Sterile Insect Technique**

  Sterile insect technique(SIT), is one of the earliest applications of genetic biocontrol. It involves using gamma radiation sufficient to cause infertility in large number of individuals, who are subsequently released. In some species, such as the screw worm *Cochliomyia hominivorax* the use of gamma dosage sufficient to cause chromosomal

breaks in the germ line (causing complete infertility) was not sufficient to substantially reduce competitive fitness, causing a highly effective suppression of the screw worm - eradicating it in the southeastern United States[33].

However, not all species respond as well. *Aedes aegypti* mosquitoes experienced a significant reduction in fitness, and therefore an insufficient sterile population[21].

Although SIT has been a successful approach to control some insect pests, there are disadvantages associated with its use.

1. A dedicated facility must be created to mass-irradiate the animals. This requires radioactive sources and is an expensive and technical process. While not the same as nuclear power, in NZ there is always push back against radioactive technologies being implemented.
2. An excessive number of irradiated individuals must be released into the wild, to overwhelm the natural breeding process, in this window of high pest population, irreparable damage may be inflicted.
3. Some species are not suitable for this approach due to impossibilities in rearing and control, slow breeding cycles, or their particular breeding systems - e.g. certain species mate with males constantly, storing and releasing sperm as needed to become pregnant, so only one fertile mate is needed in the population[44].

- **YY Males**

  Hamilton is credited with a proposition that the use of males with YY chromosomes could completely shift the sex ratio of a species of a single sex. An application of this concept, termed the Trojan Y Chromosome (TYC) approach was formally explored first in a mathematical model evaluating the potential of the method for eradicating an invasive Nile Tilapia Oreochromis niloticus (L.) population[25]. In this approach, certain tharpies may be used (usually hormonal) to produce egg producing fish with two Y chromosomes (which makes them egg producing, femenized, biologically male

individuals) When a YY (feminized) male mates with an XY male, their offspring are all males, either XY or YY. When an XX female mates with a YY(non-femenized) male, their offspring are all XY males. As such, the population will shift to an entirely male population, and then collapse.

This eradication techinque does not use genetic engineering, only hormonal. As such, it can be very successful when widely implemented, especially if immediate solutions are not a concern (as the process takes at least a few generations to collapse a population). This model is currently being utilized in the US, and is the only such genetic bio-control being utilized. However, the requirements of needing a hormonally sensitive species that can be feminized and the continual introduction of YY males have significant up front and short term costs, both in terms of dollars and environmentally[27].

- **Trojan Female**

  The Trojan Female Technique (TFT), is a twist on the SIT approach. All offspring of a mating have some DNA from either parents, however, offspring's mitochondria are inherited only from the mother. As such, the mitochondrialDNA(mtDNA) can be used to carry specific mutations that reduce fitness. If this fitness reduction only occurs in males of the species, then the number of females bearing the mutation increases, while the number of males (overall) decreases[24]. A variety of naturally occurring mtDNA mutations that reduce male, but not female, fertility have now been identified in fruit fly and hare populations[47].

  However the existence of these mutations in other species have not been extensively investigated. Modelling TFT has shown that under a wide range of conditions effective pest control may be achieved, either through few small repeat releases of mutated mtDNA into the population or single large releases (10 percent of the population)[24].

  Although this technique shows promise as a species-specific, reversible, and humane form of population control there is very little empirical evidence, such as only %8

reduction in fruit flies across 10 generations[69].

- **Gene Drive**

  Gene drives are genetic elements with biased inheritance, meaning that they are more likely to be passed on to the offspring of a pairing. This is not the same as elements which are dominant (which have the same chance of being passed on, but are expressed over top of recessive genes). At the molecular level, a synthetic gene drive consists of an expression cassette encoding a site-specific endonuclease. Tmportantly, this cassette is inserted into a chromosome at the genomic site that is cut by the endonuclease (e.g. the CRISPR/Cas9 system). Without getting too into depth on CRISPR/Cas9, there are a number of possible uses, such as shredding genes (thereby converting XY males into X0 females that produce infertile offspring males). This technique has consierable ability to control pest species[60].

  While naturally-occurring gene drives have been identified (e.g., T allele in Mus musculus L.), the recent advent of CRISPR/Cas9 gene editing technology has enabled generation of synthetic gene drives that in theory could be adapted for use in any sexually reproducing species[22].

  Although gene drives, and especially CRISPR/Cas9, are powerful solutions they are not without risks. Notably, their containment and testing are potentially very risky. The escape of these modified genotypes from their targeted populations is termed "transgene escape". There are two methods which cause this escape; the first of these occurs at the spatial level in which the gene drive could move into a non-target population of the target species, termed "intra-specific transgene escape" [28]. While not a concern in New Zealand where "every individual rat is a pest, no matter where they live" this may be a concern where populations can move offshore, such as migrating pest birds, or where the species is of particular vulnerability in other places, such as brushtail possums, which are a pest in New Zealand, but endangered in Autralia. Secondly, the

gene drive could move into a closely related species at the release site (inter-specific transgene escape), for example moving from the pest species (rats) into a protected species that is genetically similar and protected (bats).

2.6 Effects of Species Eradication

These efforts are vital and are the greatest hope for recovery of many endangered native species of flora and fauna. However, the process of eradication can be highly expensive and incredibly politicized. Further, there can be substantial risks, both in terms of lack of efficacy, and unintended side effects. Further, a large scale and sudden change in the environment almost certainly will have unexpected consequences on the other species connected to the primary target. While many of these consequences may be desirable or even intended (increased population of prey species, for example), there may also be undesirable consequences.

The complexity of food webs is well known, and removing a species is not as simple as snipping it out of the web. Unintended consequences may occur for a number of reasons when a species (the target of the control) has unexpected functional roles in food webs, such as suppression of other predatory species (such as feral cats eating rats in NZ), providing habitat creation (possum fur providing nesting materials, or pigs and deer providing clearings and forest-edge habitats), or support ecological processes that are important to native species (such as possums aiding in breaking down leaf-matter for invertebrates)[55]. These food web interactions may have outcomes that increase rates of predation upon native species, and perhaps also reducing the availability of resources to those same species[72]. Further, there are biophysical impacts related to net population mechanics which are unrelated to predator/prey interactions; pollination, seed dispersal, nutrient transfer, speed of resource recycling, and sediment stability[72],[37].

In some cases where the outcomes of species control are not considered carefully, the control of invasive species can lead to ecosystem degradation, requiring a second phase of

habitat restoration after the target species has been eradicated[7]. Large scale eradication of *any* species must be considered very carefully and a view of the entire ecosystem must be considered, however this is largely outside the scope of this project.

One prime example of species control having drastically negative outcomes can be seen on California's Channel Islands, and paints an example of how vitally important consideration of whole ecosystems is important. Eradication of feral pigs *Sus scrofa* created a change in the ecosystem whereby the island fox *Urocyon littoralis*, a critically endangered species on the island, had its survival threatened[13]. Feral piglets attracted golden eagles *Aquila chrysaetos* to the island, which prey on both pigs and native foxes. DDT becamse widely used in the 1960s, and it is likely that bald eagles *Haliaeetus leucocephalus* were able to exclude golden eagles from the island. Bald Eagles prey on fish and carrion but not on foxes. The removal of feral pigs caused increased predating on foxes by golden eagles, due to the reduction of bald eagles. As such, after feral pig control, golden eagles had to be translocated off the islands, and bald eagles reintroduced[8]. Subsequently, the fox population has dramatically increased, which shows the great potential for good to come from species control; however, simultaneously show the clear impact of the removal of an ecologically influential invasive species and the mitigation measures that must be managed following its eradication[34].

3 Species Classification and Quantification using Deep Neural Network Architectures

3.1 Problems and Possible Solutions to current monitoring techniques

Problem	**Proposed Solution**
Human observers are unreliable	Normalize/ Average the human analysis or remove it entirely
Human observation may include many hours of waiting	Expose human experts only to the observable/ relevant moments
Animals do not always interact appropriately with tools	Remove animal interaction as a requirement for observation/recording
Tool may only be able to observe one particular species	use a more adaptable/species independent tool
There is too much uncertainty as to what time an observation has occurred	Record the exact time an animal has been observed.

The proposed solution is one that addresses each of the problems identified in as shown in Table **??**; the use of a convolutional neural network which is capable of performing species-level classification on thermal footage. This would allow a real-time, reliable analysis of animal footage, without requiring human interaction, trap interaction by the observed animal, or (after training) any large time investment for analysis.

3.2 Pattern matching and species classification with machine learning

Image Categorization is the process of using some sort of visual input, such as video or still images, to identify particular objects. Historically, this process has been done using simple pattern matching - comparing the image to a known library, and attempting to control for things like rotation, scale, skew, etc. However, a more modern approach involves a deep learning based approach, whereby a deep learning convolutional neural network is "trained" on a large number of images, and essentially learns the patterns and classifications.

In a simple pattern matching program, an animal may be recognized using simple patterns, E.G. "four legs", "a long tail", and "pointed ears". However, this approach fails when the image is cropped, obscured/occluded, or for a-typical examples (such as a cat with one ear). Similarly, there is a limit in that the variations between animals can quickly become computationally intractable, and therefore attempting to design enough patterns/filters to recognise the differences by hand is infeasible.

Neural networks offer an approach to solve the problems associated with pattern matching by avoiding explicitly labelling patterns and matching techniques and instead using training. Given a training dataset, neural networks can learn on their own, essentially defining their own "patterns" of interest. A deep learning neural network is a large network layers of neurons, each of which is a way of modelling a given mathematical function. Deep learning means that many layers of classification are stacked on top of each other, with each layer increasing in complexity and therefore identification power.

The field of Deep Learning is still relatively new, having only been started really in the 2000's by Igor Aizenberg and colleagues in the form of boolean threshold neurons [2]. The field is maturing quickly and there are a number of problems being tackled, and image recognition is only one of them.

The specific implementation of the neural network is referred to as its "architecture", and the particular architecture of a neural network defines how sensitive and accurate it is,

how quickly it is able to process images, how large it is, and how much processing power it requires to train and classify. Generally speaking a framework is a named and trained implementation of an architecture.

3.3 Deep Learning Network Architecture

- **Filters**: Each filter is spatially small (within an image) in 2 dimensions (length and width), but extends through the full depth of the input volume, a typical filter may be written as "5x5x3", i.e. 5 pixels wide, 5 pixels high, and 3 pixels deep. The filter is "convolved" across the width and height of the input volume, and a dot product is calculated at each point between the filter and the input. This generates a 2 dimensional "activation map"[56].

- **Convolutional Layers**: The convolutional layer is the main processing component of the convolutional network, as such it does most of the computational work. The convolutional layer's parameters consist of a set of filters. Its output is a number of 2 dimensional activation maps equal to the number of filters used. The network will learn filters that activate when they see a specific feature (such as a straight line, or a particular color) on the initial layers, and eventually patterns (like leaves or a face) on the successive levels.

- **Pooling**: Convolutional Networks may include local or global pooling layers. The pooling layers simply combine neuron clusters in one layer, into a single neuron in the next layer. There are a number of ways to combine neuron clusters, for example *max pooling* uses the maximum value from each cluster of neurons in the previous layer[35]. Although max pooling is the most common, there are other functions such as average pooling, l2-norm pooling, and region of interest pooling[54]. Region of interest pooling is particularly useful as it specialises in having multiple objects in the same image.

- **Fully Connected Layers**: Fully connected layers are layers where every neuron in

one layer connects to every neuron in another layer. In general these layers are power/processing intensive, but are highly sensitive and accurate, and therefore are usually used for the final layers, after reductions have occurred.

- **Dropout Layers**: Dropout layers are a regularization method to reduce overfitting. Overfitting is an analysis that corresponds so closely to the data that it fails to fit additional data, or predict future observations, and is usually caused by having more parameters than are justifiable. In machine learning, overfitting and underfitting are both problems, and are usually referred to overtraining and undertraining. A dropout layer is a way to cope with the problem of a neural network having too many parameters. The main idea is to randomly drop neurons (and their connections) from the neural network during training. This prevents neurons from co-adapting.[61, 66, 5, 3]

- **Activation Function**: This is simply a boolean function that is either on "on" or "off" depending on the input. The activation level is based on some threshold that must be met for the activation function to return true. There are a great number of activation functions, but the most widely used used one is the rectified linear unit (ReLU). The ReLU function is f(x) = x+ = max(0,x), where x is the input to a neuron. this function was first expressed in a biology paper in Nature[26], and has since been used as the preferred activation method in convolutional neural networks.[38, 49].

3.4 Popular Frameworks

There are a number of frameworks that are successful, each of which have their own particular structures and designs. The choice of the "best" framework will be a part of the investigation of this project, and then tweaking some of the key parts of the framework will help to get the best results for the lowest power and processing cost, using both the visual information as well as temporal(motion)information. A few of the relevant frameworks have been detailed below.

- **AlexNet**[35] is a deep CNN. The network architecture is relatively simple compared to more modern implementations. Architecture: The network has 5 convolutional layers, max pooling layers, dropout layers, and 3 fully connected layers at the end. The filters were sized 11x11. In order to decrease training time, ReLU was used over tanh functions. AlexNet also used dropout layers to prevent overtraining. As an example it took 5-6days to train the machine using 15 million images.

- **ZF Net**[74] is heavily inspired by the AlexNet model. Architecture: ZF is very similar to AlexNet, except that ZF used 7x7 sized filters where Alexnet used 11x11. The reason for this is that using bigger filters, a lot of pixel information was being lost. Another useful aspect of the paper and implementation is that it includes deconvolution networks. This can be used to see which image pixels pass through each filter and helps provide insight into how CNNs work. This took 6 days to train on a GPU using 1.3 million images.

- **VGG Net**[59] is again based on the AlexNet type architecture. it uses 3x3 filters, however. Furthermore, the number of filters doubles after every max pooling operation. The authors state "having 2 consecutive 3x3 filters gives an effective receptive field of 5x5, and 3-3x3 filters give a receptive field of 7x7 filters, however using this technique far less hyper-parameters need to be trained in the network. VGG net was trained on 4 GPUs for 2 to 3 weeks.

- **GoogLeNet**[63] uses "inception modules" as shown in Figure 3 which are components of the larger neural network that include pooling, convolutions, and filtering, in a way that is self contained and allows for stacking, or linking many of these modules. The google architecture is 22 layers deep, with 5 layers of pooling. GoogLeNet uses 9 of these modules, and it eliminates all fully connected layers. It also uses average pooling to from 7x7x1024 to 1x1x1024, which saves a lot of parameters. Training time took less than a week with some high end GPUs.

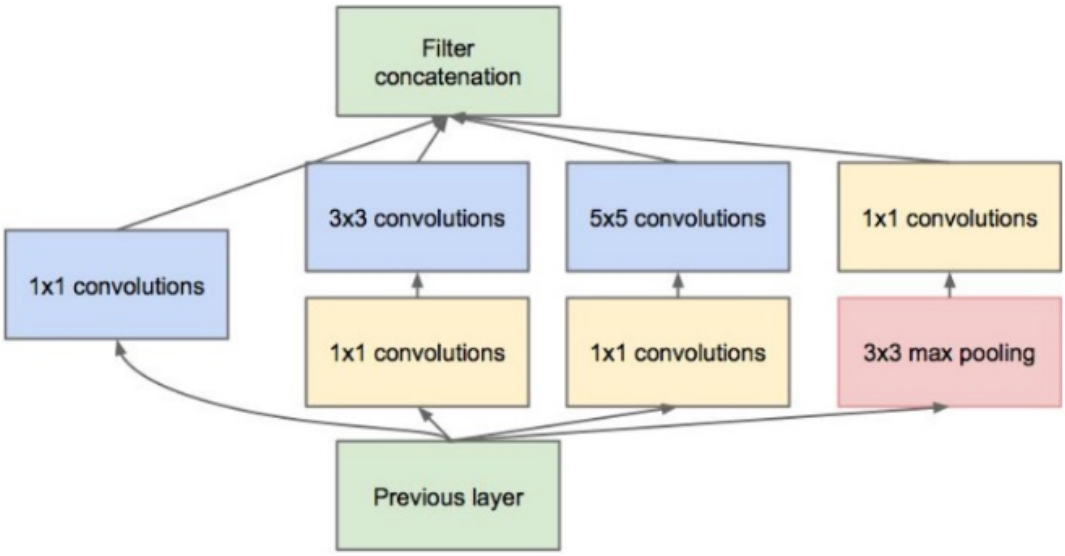


Figure 3: Google Inception Module.

- **Microsoft ResNet**[29] has 152 layers. The authors also showed that if you continually add layers the error will keep decreasing. This is in direct contrast to "plain nets" where adding more layers results in higher training times and testing errors.

  ResNet took two to three weeks to train, on an 8 GPU machine. The reason that a residual neural network is able to skip over layers using skip connections or shortcuts. While initially only able to skip a single layer of processing, with an additional weight matrix to skip weights it is referred to as it is referred to as a highway net. By skipping over layers, it is possible to have such a deep network (over 5 times deeper than GoogLeNet), while still increasing performance, and not being massively processing heavy.

4 Deep neural network architectures on thermal images

The primary body of work for this research is on the implementation and assessment of a neural network able to classify and count pest species in New Zealand. As such, the main focus of the neural network falls under video classification. There is a large amount of research already published on the topic of video classification, as well as various proposed solutions. However this is still a very indeterminate problem, and the state-of-the-art framework is not yet agreed upon. Furthermore, there are relatively few frameworks focused on thermal imaging video classification - and seemingly none that focus on *animal classification using thermal imaging video.* A few exemplary prior research approaches have been detailed below, and while none directly deal with animal detection from thermal video, they will all help inform the framework to be used in this project.

4.1 Pedestrian detection in thermal images using saliency maps

This paper uses a simple binary classification on extracted features (candidate pedestrians).

An algorithm for IR image pedestrian detection is proposed in this paper, using fuzzy C-means clustering. This paper also uses a CNN after candidate pedestrians are identified, and human posture characteristics are used to find the centroid of the figure [31].

This paper is highly relevant to this project because of the use of thermal imaging for classification, however an obviously critical feature missing from this paper is multiclass classification. Furthermore, this paper uses active IR imaging, not passive thermal imaging. Their approach to segmentation, however, is quite relevant, further, in the future if the centre of an animal is needed to be found, then the approach of using posture detection to identify it is a good approach.

4.2 Anomaly detection in thermal images using deep neural networks

This paper uses detects temperature anomolies by comparing temperatures of equipment to reference temperatures.

In this paper, an automatic method to detect thermal anomalies based on deep neural networks (DNNs) is proposed. The DNN model is trained to learn the statistical regularities of normal thermal images, and anomalies are detected based on pixel-wise comparison between the learned reference temperatures and the measured temperatures[40].

Essentially this paper is detecting anomalous operation using a DNN. This paper is somewhat relevant to this project due to the use of novelty detection, however there are a few limitations, notably again it is a binary detection of difference, and again it uses active IR and not passive thermal imaging.

4.3 Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure

This paper uses a thermographic camera embedded in a UAV to gather footage of solar panels. Two novel region-based convolutional neural networks are unified,using the combination of thermography and telemetry allows for a fairy robust detection structure, creating a location and status of the solar panels [30].

This paper is very relevant to the project due to its use of passive thermographic cameras to detect hot spots, which is very similar to the proposed use of thermographic cameras to detect animals (where animals can be assumed to be warmer than their surroundings). However, the self proclaimed main contribution of this paper is the use of telemetry (which will not be relevant to this project, as all cameras will be in fixed locations) as well as, again, the lack of multi-class classification.

4.4 Intelligent Fault Diagnosis of Rotor-Bearing System Under Varying Working Conditions With Modified Transfer Convolutional Neural Network and Thermal Images

In this article, a framework for fault diagnosis is proposed.

1. Thermal images are collected and pre-processed
2. To overcome classical training problems, a modified CNN is developed using stochastic pooling and rectified linear units.
3. Parameter transfer is used to adapt the CNN to the target domain[57].

The use of thermographic imagery and transfer learning make this paper highly relevant to this project, because of the lack of a good dataset.

4.5 Fully automated DCNN-based thermal images annotation using neural network pretrained on RGB data

As noted above, one of the biggest challenges of training deep neural networks is the need for massive amounts of data. However, currently, there are no large-scale thermal image datasets that could be used to train the state of the art neural networks, while there are bountiful RGB datasets available.

This paper presents a method to map RGB labelled data onto thermal imagery. This means that the accuracy of RGB trained nets can be transferred onto thermal imagery. The method proposed in this paper uses an RGB camera, a Thermal camera, and a 3d LiDAR.

The paper itself is incredibly relevant as an early identification of difficulty in this project would be the lack of data. Potentially using a well made RGB dataset to help train a thermal imaging-based net would lead to great increases in accuracy while on a relatively small dataset. However, there are some limitations why this approach cannot be used: LiDAR

does not play well with moving objects, and a focus of this project is on video/real time classifications. Furthermore, this technique requires both an RGB and IR image of the same objects, which requires the footage to be taken in the light. However, It's still possible that the use of this dataset could essentially be used to pre-train the network, before presenting any animals to be classified. However, due to time constraints (and the success with gathering crowd labelled data), this approach was not used in the final development of the project[39].

Cacophony.org website

Figure 4: Infrared Illuminated Possum

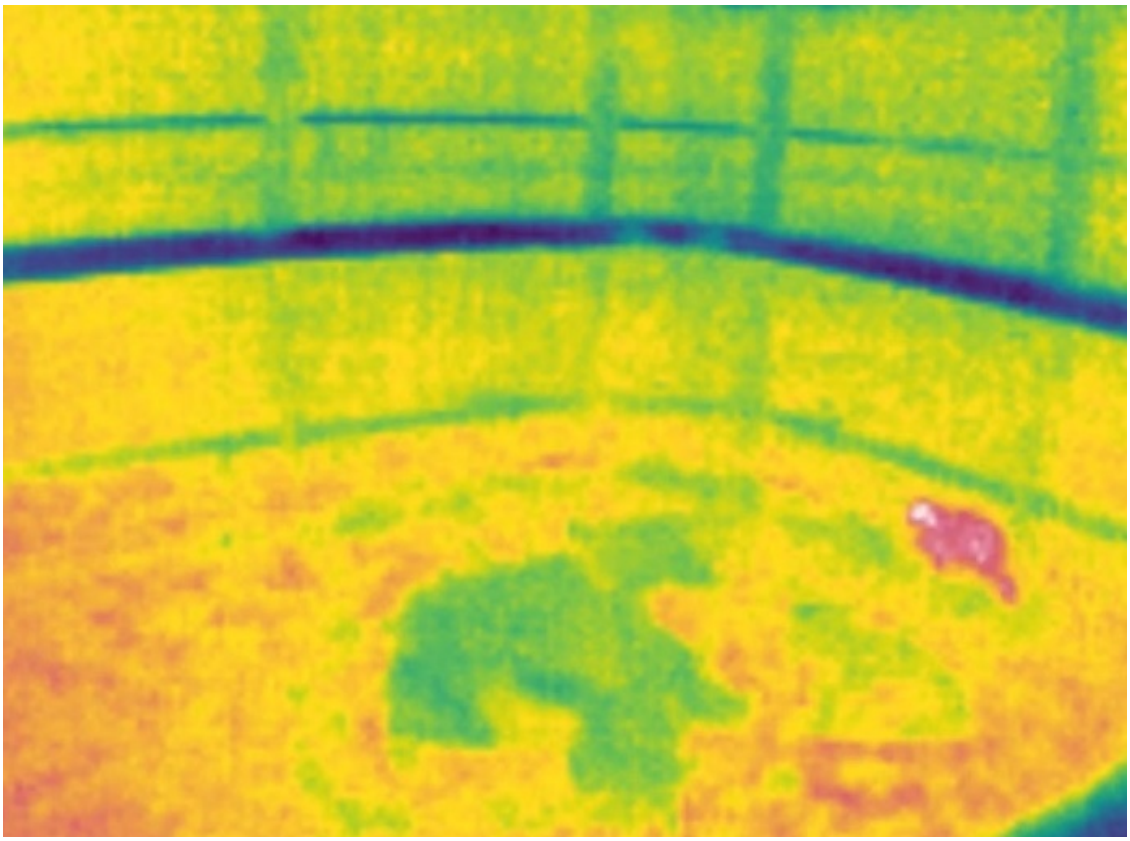
5 Proposed Method and Implementation

5.1 Environmental Constraints Working in New Zealand

Many of the pests in New Zealand are crepuscular or nocturnal[68, 11]. Because of the low light levels using a standard camera is impossible. In most cases where low light levels are a factor, an I2 (Infrared Illuminated) camera is used as it is not light dependent and can offer a comparable resolution as shown in Figure 4. I2 cameras try to generate their own reflected light by projecting a beam of near-infrared energy that their sensor can see when it bounces off an object. This works to a point, but I2 cameras still rely on reflected light to make an image, so they have the same limitations as any other night vision camera that depends on

reflected light energy – short range, and poor contrast.

While these cameras are generally a cheap and effective way to film animals at night, their use for this project is impossible due to the sensitivity of some pest species to IR lights, such as possums. Because of this some of the pests to be monitored specifically avoid IR cameras, due to having an IR spotlight shone on them[43, 64].



Cacophony.org website

Figure 5: Cacophony Thermal Camera

5.2 Recording Technologies

In addition to the above surveying techniques, there is also the option of filming and photographing animals in the NZ bush. There are a few different ways this data can be reasonably gathered. The choice of which camera technology to use is vital to the success of this project.

- **RGB or Daylight Cameras** use the light of the sun or other visible-wavelength illumination to see. The light from the source bounces off objects and into the sensor. The obvious upside of these cameras is that they emulate human vision and are thus relatively easy to calibrate, set up, and trouble shoot. However they have also numerous drawbacks, such as operating in the same spectrum as many animal's camouflage, and requiring light (in this projects case, either a remote light or sunlight would be required).

- **Night Vision Goggles** take minute amounts of light and massively multiply their luminosity through digital means. Cameras may apply this same technique to achieve the same effect. However, this technique has a number of limitations; firstly there must be SOME visible light - the wavelength viewed is only the visible spectrum of light. Similarly, when amplified, the wavelength is essentially normalised, and so light is shifted towards the middle of the visible spectrum, which gives this technology its classic "green filter" appearance. If there is no light, or very limited light, then the amplification process rapidly fails due to the signal-noise ratio becoming overwhelming. Though generally star-light alone is plenty for this technology, when also filtered through the canopy or an overcast night, this technology does not have enough light to amplify and cannot see well. Furthermore, this technology does not work at all when there is too-much light, and the technology must be turned off or risk dazzling the camera/viewer.

- **Infrared Illuminated Cameras I2 Cameras and Active Infrared Cameras** are all names for cameras which project an infrared spotlight, and have lenses capable of seeing the infrared light. An example output of these cameras can be seen in Fig 4. Two advantages of these cameras over NVG cameras are that they do not require any natural light at all, as they produce their own "spotlight". Further, because they do not detect visible light, they cannot be dazzled by most lights, even the setting

sun does not produce very much infrared light, as it primarily emits from the visible spectrum and above (into ultraviolet). However, some animals are able to see/detect infrared lights, and some of these animals are of interest, such as possums, which have been observed to avoid areas illuminated by only infrared light, which is invisible to humans.

- **Passive Thermal Cameras, Heat cameras, Thermal Cameras, and Thermographic cameras** are all names for cameras which detect the *long wave infrared radiation* emitted by the NZ species of interest in this research. The exact method by which these cameras work is outside the scope of this project, but it is sufficient to understand that all matter above 0 degrees Kelvin emits infrared radiation(also known as black-body radiation) as they cool. This energy is emitted as *long wave infrared radiation*, with the warmer objects emitting more infrared radiation. Photons also emit *shortwave infrared radiation* when they bounce off objects, however this of course requires a lightsource to be present. Thermographic cameras are able to measure and amplify the differences in long wave infrared radiation, making warmer objects stand out from their background. An example output of these cameras can be seen in Fig 5. Both Infrared Illuminated Cameras and Passive Thermal Cameras use wavelengths of light in the infrared spectrum, but the critical difference between them is that infrared illuminated cameras project a spotlight using infrared LEDs, while passive thermal cameras rely on the constant emission of radiation by all warm objects (i.e. living creatures).

  All objects emit passive infrared energy, called blackbody radiation, as a function of their temperature. Furthermore, Animals emit much more radiation than their surroundings (usually trees and plants). This radiation is emitted as far infrared radiation, and is invisible to most animals, including the pest species in New Zealand.

  Thermographic cameras are designed to detect this radiation in the same way as regular

cameras detect visible wavelengths. These cameras are also capable of producing an image in total darkness, as the objects themselves emit the detected radiation as shown in Figure 5.

High Resolution thermographic cameras, however, are much more expensive. Because normal glass blocks far infrared radiation, lenses in these cameras must be made from more delicate or expensive materials such as sapphire or germanium. Furthermore, images from these cameras also tend to be monochromatic, along a single axis of color, as attempting to reconstruct colour from infrared radiation is extremely complex, and likely not all that relevant for this application, where most of the animals in question are similarly coloured.

As such, in order to get a very high resolution image the camera would be too expensive, as well as power hungry, both of which are not suitable for a low-maintenance widespread monitoring solution placed in the wilds of New Zealand. As such the identification system must use a fairly limited amount of information from affordable, very low resolution thermal images to enable its differentiations.

5.3 Limited Onboard Processing Power

The end goal of the project is to perform on-site identification of species. Additionally, because the tool is to be placed in many locations within the New Zealand bush, this device must be relatively low-cost, rugged, and run on limited battery life-span (to minimise replacement/recharge factors). The logical choice is a small, single board computer such as a Raspberry Pi, the obvious drawback of which is having very limited processing power.

5.4 Data Storage Issues with Long Surveillance Time

The goal includes being able to record animal visits throughout the night, as such the camera must be monitoring and watching for visits for approximately 12 hours every night. Due to

storage costs, it is infeasible to record 24/7 footage in a useful resolution, so the Raspberry Pi and on board camera must have a certain level of logical processing that stores only the footage that may be of interest. This does not have to be perfect, as the cost of a false positive (storing footage with no animal present) is minimal (it will be further processed, and then discarded by the neural network), but the cost of a false negative (i.e. not storing footage of an animal visit) is the opportunity cost of lost training data (and later lost identification of pests).

This is of particular concern if there is a systemic false negative, such as not being able to record when there is a cat, even if it works for possums and rats. This type of systemic false negative would result in not having any training data at all for a particular species or environment.

5.5 Dataset Generation

One key difficulty of training neural networks is the gathering of large volumes of data, that have been accurately labelled (usually via crowd sourcing, expert analysis, or many hours of work). In other projects or areas of research this problem can be avoided by using a dataset that has already been labelled, such as the COCO dataset, which includes over 200,000 labelled objects in context[12][41] as shown in Figure 6. These large datasets are excellent for training neural networks because they have the following features, all of which are essential:

- There are images of the object from multiple angles, allowing the net to still accurate identify the object as it moves through relative space
- The labelled images are sometimes partially occluded, which allows the net to still identify objects as they pass in front and behind one another
- The labelled images occur in many different contexts, such as a car in a forest vs on a street, and a car at night vs a car during the day. This allows the net to more accurate

gauge the "important" part of the analysis.

- The labelled images are segmented, separating the "object" from the "background" (even when there are multiple objects, or when the object is in the background and the foreground is the part being segmented)

However, these datasets cannot be used for this research, because no classes exist in it for low resolution thermal images of animals at night. Therefore a contribution of this research is to create a dataset that also possesses the above classes. In classic discriminative examples such as cat versus dog, the image recognition software must overcome issues of viewpoint, lighting, occlusion, background, scale, and more. Data Augmentation may be used to bake these translational invariances into the dataset such that the resulting models will perform well despite these challenges [58].

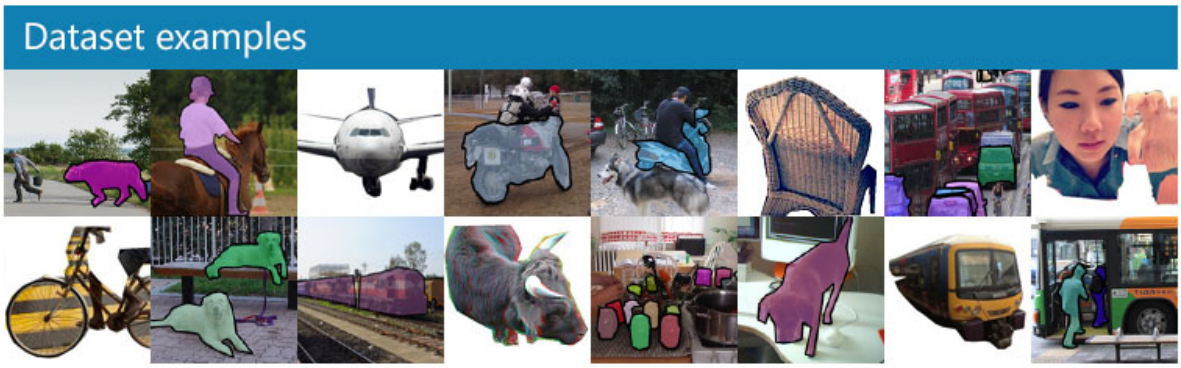


Figure 6: COCO Dataset

Having a large and labelled dataset can be helpful, not just for saving time, but also for accuracy. It is vital that the training data is a true representation of the total dataset. The more data the net has access to, the more accurate it is.

Further, the use of thermal imagery means that other infrared datasets such as the ADAS dataset[23] cannot be used. The FLIR dataset is used for ADAS, or Advanced driver-assistance systems, and as such is exclusively urban footage and contains labelled images as shown in Figure 7.

Figure 7: FLIR Dataset

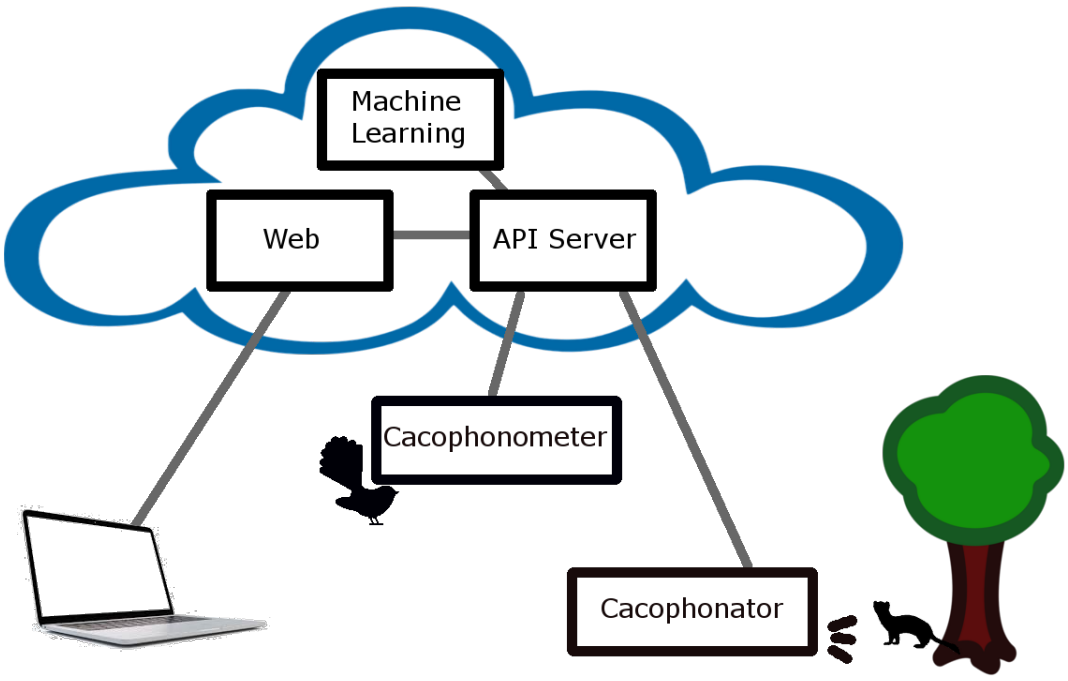
The thermal camera will record a full night of video, and store it on a local hard drive. The hard drive will then be taken to a server in order to be processed into a form that the network can be trained on. The footage will be checked for the binary result; "contains region of interest" vs "does not". The footage that does not contain any animal or hot spots will be discarded. Then the footage will be split into individual occurrences of animals, attempting to make each occurrence a single "event", i.e. in a night of footage, there may be three "events" of animals appearing, and the remaining footage when there are no animals present will be discarded. The footage not discarded will then have to be labeled. An overview of the system can be seen in Figure 8.

5.6 Crowd Sourced Data Labellign

Due to the problems discussed above, all of the data must be gathered by passive thermographic cameras (using the specific camera setups), and labelling large amounts of data will likely require crowd sourcing, with quality checks to ensure the data remains accurate.

The approach for data labelling will be to modify the footage shown to users, using the following steps

1. Minimise the amount of erroneous data pragmatically - this will include as many false



Cacophony.org website

Figure 8: Cacophony System Diagram

positives, false negatives, and empty footage.

2. Minimise the amount of individual decisions/clicks the user must perform to label a piece of data, to minimise fatigue and maximise useful labelling

3. Add redundancy and suggestions to help with accuracy.

This process will be ongoing throughout the project, but once there are a few hundred labeled segments the network can begin being trained.

5.7 Training the neural network

Some of the recordings will then be used to train the neural network, and some used for validation. The validation data is used as a proxy for novel data, and so performance can be better predicted by having a better validation dataset. However, the model should never be

trained on the same data that appears in the validation set, and the more unique data that appears in the training set, the more reliable and accurate the model itself can be. There are a number of different ways to split the data into training sets and test sets for cross validation of the model, such as the following.

Leave p out: this method is exhaustive (meaning every possible split is tried) and involves using p observations as the validation set, and the remainder as the training set. This is then repeated for all ways to split the data. Performing an exhaustive method on a large data set is computationally infeasible, however.

X:Y or holdout: This method involves simply randomly splitting the data into two groups for training and validation, for example 90:10, or 50:50. This is a simple way validate data, but is susceptible to chance, such as all data of a certain type being found in one group (e.g.,all images of possums appear in the validation set)

K-fold: this method involves splitting the data into k random subsets, using k-1 sets for training, and the remaining subset for validation. This is then repeated using identical subsets until all k subsets have been used for validation. K is commonly set to 10, but can be any value. When k=number of observations, it is identical to leave one out validation.

Monte Carlo: the data is randomly assigned to training or validation. The model is then fit to the training set and verified by the validation set. This process is repeated an arbitrary number of times, and the accuracy is the average over all splits. This has an advantage over k-fold in that the proportion of training:validation is not determined by the number of folds. However, it is susceptible to chance; some data may never be selected, while others may appear more than once. Furthermore this method of validation is susceptible to Monte Carlo variation, meaning that the results are not deterministic - they will change every time the data is trained.

This will be an iterative process, whereby the network will be trained, verified, and tested, before being tweaked and updated to deal with novel cases that throw errors, such as two animals being seen at once, obscured animals, animals that move too close to the camera,

etc. There will almost certainly be edge cases that cannot be recognised as problems without iterative improvement removing the larger errors.

For this project, leave p out is impossible because of the size of the dataset. K fold is the preferred method, as it approximates leave p out, while requiring less training time. However, if the training time of the model becomes too long, it may become infeasible to use k fold. If that is the case, the holdout method will be required.

5.8 Network design and Architecture

Most of the existing frameworks are designed to work with generally high contrast, static images of a great many different possible classes (in the thousands).

The network for this project would have much lower quality photos, but also fewer classes (in the tens), and additionally have access to motion information. An additional complication that must be overcome with this novel framework is temporary occlusion due to foliage obscuring the animal.

The network will be a convolutional neural network, which allows multiple layers and more complex decisions to be made. The increased complexity in decision making will allow for better results given the low quality images. Because the images are low resolution motion will also be used, by using the flow of the identified region of interest over time. The motion of animals can be used in the general sense to inform the network - i.e. "hedgehogs do not climb trees". Additionally, by using the knowledge that an object in frame X is the same object as that in frame Y, choices can be made, for example catering to partially obscured animals, or an animal changing the direction it is facing.

Pre-training a neural network, and "trimming" off the final few decision making layers can allow for a faster training process by skipping a lot of the training epochs, while also providing good results by reusing "known good" neurons. However, the effectiveness of pre-training diminishes as the differences between the dataset that the net was originally trained on and the dataset that it will now be used on increase. Because the dataset that

this research is being applied to is unique (i.e. thermal video data), pre-training will not be used.

5.9 Data acquisition

For reasons detailed above, thermal cameras attached to small Raspberry Pis uploading their video capture data to a web server is the method we've chosen to acquire the raw video footage. There are currently 45 of these meters placed in and around the New Zealand native bush. There are only 45 because the meters are constantly being iteratively improved, however their placement (and therefore proportion of different animal visits overnight, as well as angles and backgrounds) can be varied regularly.

These devices wake up for a certain period of time (overnight, from the hours of 6PM to 8AM) and record footage using their thermal camera. During the recording window the camera is always on and I run a simple temperature filter and motion detection algorithm over the incoming footage. If something warm and moving is detected then start recording to disk. Recording continues until there's no more motion detected (up to limit of 10 minutes). This recording is then uploaded to the Cacophony Project API server for storage and labeling.

The camera itself uses a waterproof thermal lens, a power bank, with a Raspberry Pi to control it. The footage is run through the simple process described above to only upload the footage that contains a region of interest (a potential animal) (as shown in Figure 9 and Figure 10. The full hardware and firmware can be seen in Table 2

Figure 9: Cacophony Camera

5.10 Hardware specifications

Base Hardware	Raspberry Pi 3
Thermal Camera	FLIR Lepton 3
Additional Hardware	Custom interface hat ATiny microcontroller 3g Modem USB audio Adapter Real Time Clock
Operating System	Raspbian
Key Software	Letpon 3 Thermal Recorder Thermal Uploader

Table 2: Hardware Specs

5.11 Glossary of Terms

5.12 Initial Data Processing

The footage on the server arrives in an unprocessed, raw form, including some metadata provided by the camera such as ambient temperature, camera ID, and time of day. The full

Cacophony.org website

Figure 10: Cacophonator

data set consists of 15,465 recordings, covering 18 classes. Each video has a resolution of 180x120, is captured at 9 frames per second, and contains a single channel consisting of the raw temperature readings.

The footage must then be accurately labeled in order to train the RCNN. Humans are used to label/annotate this data accurately, to feed into the RCNN for development.

Clips(10 seconds to 10 minutes long) are processed into tracks (an individual object moving) likely to represent moving objects, using the following simple assumptions

- Animals are relatively small (within the frame)
- Animals are hotter than the background
- Animals move within the frame

Clips are tagged with labels indicating which animals are in the clip. If the clip has

Name	Definition
Region of Interest	A region of interest in a frame that may or may not contain an animal.
Frame	A frame of video. May be either a frame from the clip or a frame taken from a track.
Channel	Each frame generates 5 channels which are thermal, filtered, flow_h, flow_v, and mask.
Thermal Channel	The thermal channel representing per pixel temperature.
Filtered Channel	The filtered channel, which has been background subtracted
Mask Channel	The mask channel, indicating which pixels are of interest
Optical Flow Channel	The per pixel screen space velocities.
Segment	A 3-second segment of a track.
Track	A series of regions that track an object of interest through a frame.
Clip	A 10sec-10min long video clip of recorded thermal vision.
Visit	A sequence of clips containing the same animal with only short gaps between them.

Table 3: Glossary Of Terms

been tagged with more than one animal, the CRNN currently ignores it (in the future it will be useful to identify multiple animals simultaneously, but this is currently a relatively rare edge case that is outside the feasibility of current training data). If it has a single tag, it is assumed that any tracks found are of this animal. Unfortunately, this means the process will sometimes bring up false-positives, which must be removed manually.

1. **Clip Analysis**: Firstly, check if a clip has a 'static' background, that is one which does not change much or a 'moving' background. Static clips have an estimated background calculated from the 10th percentile pixel values over the clip. Tracking is much more effective with static backgrounds.

2. **Clip Rejection**: Sometimes poor-quality clips come through the system, they are filtered out here. The following rejections can occur:

    - Clips less than or equal to 9 frames (1 second) are rejected. Most likely these are corrupted video files.
    - Min/mean temperature: if the mean or max temperature of the video is outside of normal bounds(this is a manually chosen threshold, defined by the camera and

its location) the clip is rejected.

- Temperate range: If the temperate range of the clip is too high the clip is rejected. Normally this means the camera has just turned on and is self-adjusting.

3. **Get filtered channel**: For static background clips, the filtered channel is calculated as follows

$$C_s = relu(C_t - C_b)$$

$$C_f = relu(C_s - median(C_s))$$

Where Cs is the subtracted frame, Cf is the filtered frame, Cb is the estimated background and *relu* is the function

$$f(x) = max(x, 0)$$

By applying the *relu* twice, any DC changes in temperature can be adjusted for. For example, if the scene gets hotter or colder over time these will be compensated for.

On clips that had a moving background detected the following formula is used instead.

$$C_f = relu(C_t - median(C_t) - 40)$$

This algorithm is far less effective.

4. **Mask**: A mask is created by setting a threshold value. The threshold value is taken automatically as half of the 99th percentile temperature value then bounded between and 30, 50. However, the minimum threshold level can be adjusted to be more or less sensitive.

A 5x5 Gaussian blur is applied to the thermal values, then values above the threshold are set to 1, and values below or equal are set to 0.

5. **Region of Interest Detection**: Regions of interested are found by running connected components over the mask. Regions are extended 6 pixels outside the bound box that would contain the mask pixels.

   Regions with fewer than 8 pixels are ignored, as are regions whose pixel variance (the variance of the thermal pixels deltas between this frame and the previous) is below 2.0 units.

6. **Track Matching**: Regions are matched to tracks as follows:

   Regions are compared to existing tracks and given a score based on their distance from the tracks predicted location this frame, and the relative size distance. Tracks are matched greedily to the regions so long as the distance/size differences are not too extreme.

   Any remaining regions are considered as candidates for new tracks. If the region does not overlap an existing track a new track for this region is created.

   Tracks that could not be matched to a region are marked as 'lost' and terminated if they do not require a region within 9 frames (1 second).

7. **Filter Tracks**: Tracks are assigned a score based on how much they move from their origin. How much the move in general, and the number of active pixels they contain. Tracks are ordered by this score so that the first track is the most 'interesting' track. Tracks that do not meet minimum duration, mass, or movement requirements are excluded, as to tracks that overlap other tracks too often.

   Only the 10 best tracks (within the clip) will be considered for further processing.

8. **Data Labelling**: The recordings are stored on the cacophony website, (https://browse.cacophony.org where users can tag videos. Each of the recordings are presented to multiple users,

which increases redundancy in accurate tagging, and tracks only have a "correct" tag when all users agree (when there is a disagreement, the clip remains stored and presented to more humans, but is not used for training).

Collecting labelled data in general is a UX problem, requiring a lot of human-hours of work, and could require a paper in and of itself. In order to maximise accuracy of tagging, as well as decrease fatigue of users the following UX choices have been done on the website:

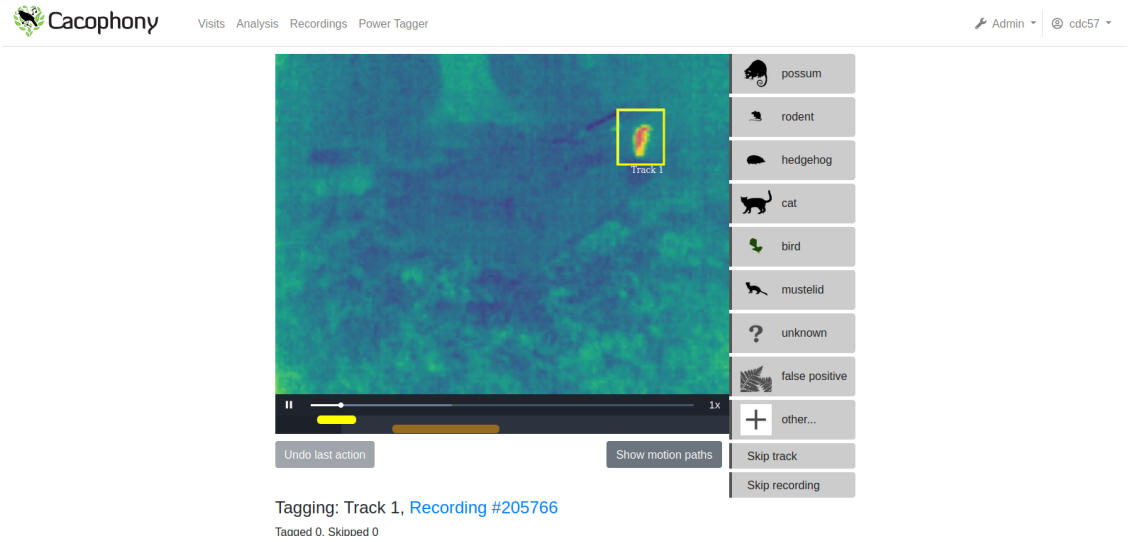
- Large buttons; a single click to label an animal (or a false positive/negative, or an unknown animal)
- The region of interest has been boxed
- The clip will loop until labelled or skipped
- When labelled or skipped, the next clip will play immediately, reducing downtime, and maintaining attention levels.

The website implementation can be seen at https://browse.cacophony.org.nz/ (an account is needed), or in Figure 11

5.13 Further Data Processing to generate trainable data for the Neural Network

1. **Generate Optical Flow**: Optical flow is generated per pixel over the entire clip. This process is quite slow so it is only performed if at least one good track was found. The optical flow algorithm used is the DualTV-L1 algorithm implemented in OpenCV (Designing a full optical flow generation algorithm was far outside the scope of this project).

2. **DualTV-L1**: An algorithm based on total variation (TV) regularization and the robust L1 norm in the data fidelity term. This algorithm can preserve discontinuities in



Cacophony.org website

Figure 11: Cacophony Crowd Source Tagging Website

the flow field and offers an increased robustness against illumination changes, occlusions and noise. In this work I present a novel approach to solve the TV-L1 algorithm. The method implemented method results in a very efficient numerical scheme, which is based on a dual algorithm of the TV energy and employs an efficient point-wise thresholding step [71].

3. **Apply Hints**: The maximum number of tracks to export for a specific clip can be set via the hints file. If this is set to zero the clip will not be processed at all. If the tracking algorithm is generating false-positives, or other animals have been included, this can be used to only output the first, and best, track.

4. **Export Tracks**: Tracks that were not filtered out are exported to an HDF5 track database. For each frame in the track, the region of interest is extracted with all 5 channels and written to the database.

   Although the processing above is done in 32bit float the channels are converted to the int16 format for better size and compression ratios. This conversion is done as follows. The raw thermal values are 14bit integers, so no precision is lost there, however, the

optical flow is scaled by 256.0 in order to maintain subpixel precision.

5. **Data Tailoring and Pre-processing**: The human-labeled data still needs some pre-processing before it can be handed to the CNN. This pre-processing is partly to reduce the amount of information that must be processed by the network, as well as to semi-normalize the data (i.e. removing attributes I don't care about, and keeping as much variance as possible in the attributes I do care about.

   Firstly, the footage is cropped to just include the region of interest of the animal itself. Then, the footage is scaled to 48x48 px. each "track" is then split into "segments", which are 3 second long windows. The input training data is a 48x48 retina, with a 16bit thermal channel, and two optical flow channels (horizontal motion and vertical). The optical flow is TV-L1 dense optical flow.

   Finally, the training segments are augmented in order to increase the amount of data available for training. The augmentations used are: scale / crop, brightness shift / contrast shift, and horizontal flipping.

5.14 Tuning the Motion Detector

A consistent problem being encountered was that some footage/recordings of animals were starting later than they should - i.e. footage begins with animals in the centre of the frame. Therefore, the motion detection needs to be tuned slightly for some animals.

Off-the-shelf trail cameras use a separate motion detector to trigger recording to begin, however my solution uses the same thermal camera, which is always on.

The difficulty of trying to fix false negatives in saved footage, is that there is no recorded footage to know where the errors emerge from. As such, using the dataset I already had, the trigger needed to be studied and repaired. Luckily, the device already saves footage for 10s *after* the final animal is seen. Therefore, any animal that *actually is* in the footage, must be a false negative. Reviewing this footage revealed that the animals present in this

footage were those that were either moving very slowly, such as a curious possum moving very hesitantly, or very small animals moving quickly, such as rats.

The previous setting I had used was reviewing changes over the previous 3 frames(or about 1/3 of a second of footage).

Various thresholds were tested, and the duration found to be optimal was 10s. Note that this does not change the number of frames being tested, but rather every new frame (the present frame) is compared to the one that occurred 10s previously, rather than 1/9s.

Another dependent variable that is influenced by this duration is the heat-change threshold over this time. As such this threshold also needs to be updated. Using a longer time frame allows for a higher threshold for temperature change, therefore the threshold has been changed from 30 unit difference, to a 50 unit difference. This change also aided in reducing false positives[48]. The values used for each of these thresholds can be seen in Table **??**.

Parameter	**Old value**	**New Value**
Time gap between compared images	10secs (90 frames)	1/9sec (1 frame)
Minimum temperature for pixel	3000	3000 (2900 one day)
Minimum change in pixel value	30	50
Consider all temperature changes or only warmer	Any change	Warmer only
Footage before animal detected	None	1sec

Table 4: Old vs New motion detector values

5.15 Converting footage into Data

1. **Breaks tracks into 3-second segments for training**

   Tracks can be of arbitrary length. The build process breaks these tracks into overlapping 3-second segments, where a new segment is taken every 1 second. I.e. a 4-second track would be broken into 2 segments, one containing seconds [0,1,2] and one containing seconds [1,2,3].

Segments with a low number of active pixels are removed, as it is unlikely that they contain enough information to train on.

2. **Split data into three sets; train, validation, and test.**

   The data in the database can be considered as follows:

   (a) Segments: The number of segments for each class (roughly speaking the number of seconds of footage)

   (b) Tracks: The number of tracks found for each class

   (c) Camera Days: The number of unique camera days this class was seen for. For example, 1 camera over 5 days would count as 5, and 5 cameras over 5 days would count as 25.

   For training the diversity of the dataset is important. For example, a single 10-minute track of a possum in a trap would generate 600 segments but contains less useful information than 3 30-second tracks taken from different cameras, over different days.

   Furthermore, in order for the test and validation sets to be good indicators of generality, they must contain the same type and ratio of data, but not be too hghly correlated to the training set.

3. **Data splitting**

   Assigns class-camera-days into either the training, validation or test set. That is, if an animal was seen on a given day, on a given camera, then data from that camera, on that day, cannot be used in either the validation or test sets for that class.

4. **Minimum requirements**

   Make sure to assign a minimum number of segments, tracks, and camera-days to the validation and test sets. The segments requirement makes sure three is enough data, whereas the track and camera-days requirement makes sure there is enough diversity

of data. For example, the test set cannot be made up of just footage from one camera on one day, even if that day generates enough segments.

5. **Heavy camera days**

   'Heavy' camera-days are days which have much more segments than normal. Assigning these to the test and validation sets wastes a lot of valuable data, so these are always assigned to the training set.

6. **Comparing results over time**

   Sometimes the dataset will need to be updated, and the results compared against previous results. If the training and validation sets change each time this can make comparing results impossible. Therefore a template file is used which is simply a previous dataset. Validation and test splits will be adjusted to use, as much as possible, the same camera-days for each class, as was used in the template. Sometimes additional camera-days will be added to meet new minimum requirements, or if another class has been added.

7. **Rebalance class labels via random sampling**

   Every segment in the dataset is assigned a weighting which is its retaliative likelihood of being selected. By adjusting the weights the dataset can be balanced such that each class is sampled with equal likelihood even if the classes contain different numbers of segments. Both the training and validation set use this method, whereas the test set uses subsampling at the segment level so that it can be evaluated in its entirety once built.

8. **Applies some filtering late in the process, such as which classes to include.**

   Adding and removing classes can be done at the dataset level without having to re-run the (much slower) track extraction process. This allows for changes late in the pipeline to be applied without having to rerun the track extracting. Minimum mass requirements can also be set at this point.

5.16 Data Augmentation

In order to overcome the classical problem in machine learning of limited labelled data, a group of techniques collectively known as data augmentation is used. The data augmentation algorithms discussed in this survey include geometric transformations, color space augmentations, selective filters, image combining, random occluding and deletion, feature space augmentation, adversarial training, generative adversarial networks, neural style transfer, and meta-learning.

Networks are heavily reliant on big data to avoid overfitting. Overfitting refers to the phenomenon when a network learns a function with very high variance such as to perfectly model the training data, as shown in Figure 12. By having a robust data augmentation system, there is a highly reduced rate of overfitting. At the 7th epoch, the data has minimized its testing error, however the training error still reduced at the 8th epoch and beyond. However, this is because the net is likely being trained on data it has already seen, so it is not becoming more generically accurate, but rather the net is only more accurate on the data in the test set. As such, the testing error (on images unseen) increases.

Figure 12: Overfitting Example Inflection[58]

1. **Starting Frame Jitter** The segments starting frame is jittered by +- 5 frames when loading.

2. **Translational and Rotational** The recorded segment is rotated and translated within the training window by up to 15%

3. **Random Cropping** Random crops are taken from the original tracking frame. The frames in the dataset are recorded with 6 pixels of padding. During augmentation, the left and right corners are inset independently by between 0 and 5 pixels inclusive causing random scaling, translation, and distortions. If no augmentation is applied the corners are inset by 2 pixels by default.

4. **Random Flipping** The segment is flipped horizontally 50% of the time.

5. **Random Level Adjustments** Thermal levels are randomly offset and scaled 75% of the time.

5.17 Loading the segment

The initial pre-processing is done on the dataset. First, a batch of segments is randomly sampled from the dataset. Then pre-processing is applied, which includes augmentation as outlined above. The cropped frames will then be scaled to the desired size, which is currently 48x48. The thermal channel will be referenced by the median thermal value of the original clip frame. This means that the thermal values are given values of 'how much hotter are I than the general ambient temperature'. This helps to normalise the values across multiple days, cameras, and locations.

Because the processing time to load segments can be high the segments are loaded asynchronously with separate processes. This allows the GPU to train while the segments are streamed in concurrently. Loading segments typically takes 2-3 CPU cores to keep up with the GPU training.

5.18 Data Pre-processing

Thermal: A threshold is applied to a copy of the thermal so that values below the threshold are set to the threshold level. This removes much of the background. The values are then scaled to be roughly unit norm. Not thresholding the thermal channel will often result in overfitting.

Optical Flow: The optical flow values are normalised to be roughly unit norm.

Thermal Stream The thermal channel is then based on a convolutional tower that processes the thermal channel.

Optical Flow Stream Optionally an optical flow stream can be used. This process the optical flow in parallel. Both optical flow channels are included, along with the thermal channel.

LSTM Units The output of the thermal and flow streams are concatenated together and feed as inputs into LSTM units. The LSTM units allow the model to process video and make connections over time with what was seen

Softmax classifier Finally, a SoftMax classification is performed on the output of the LSTM units. Label smoothing is applied to encourage the model not to make such strong predictions reducing the likelihood of high confidence, but wrong, outputs.

5.19 Classification

Once tracks have been extracted from the clip they are feed to the classifier, frame by frame for identification. Because the model is a recurrent neural network, sequences of any length can be feed into the model, however, results may not be optimal with short sequences. After classification, the post-processed footage can be seen as in Figure 13.

The per frame classifications are smoothed using an exponential moving average (EMA) in order to remove any transient spikes in the predictions and to discourage making strong predictions on short amounts of data. The following heuristics are also applied.

Track frames with a small number of active pixels are penalised in terms of their score

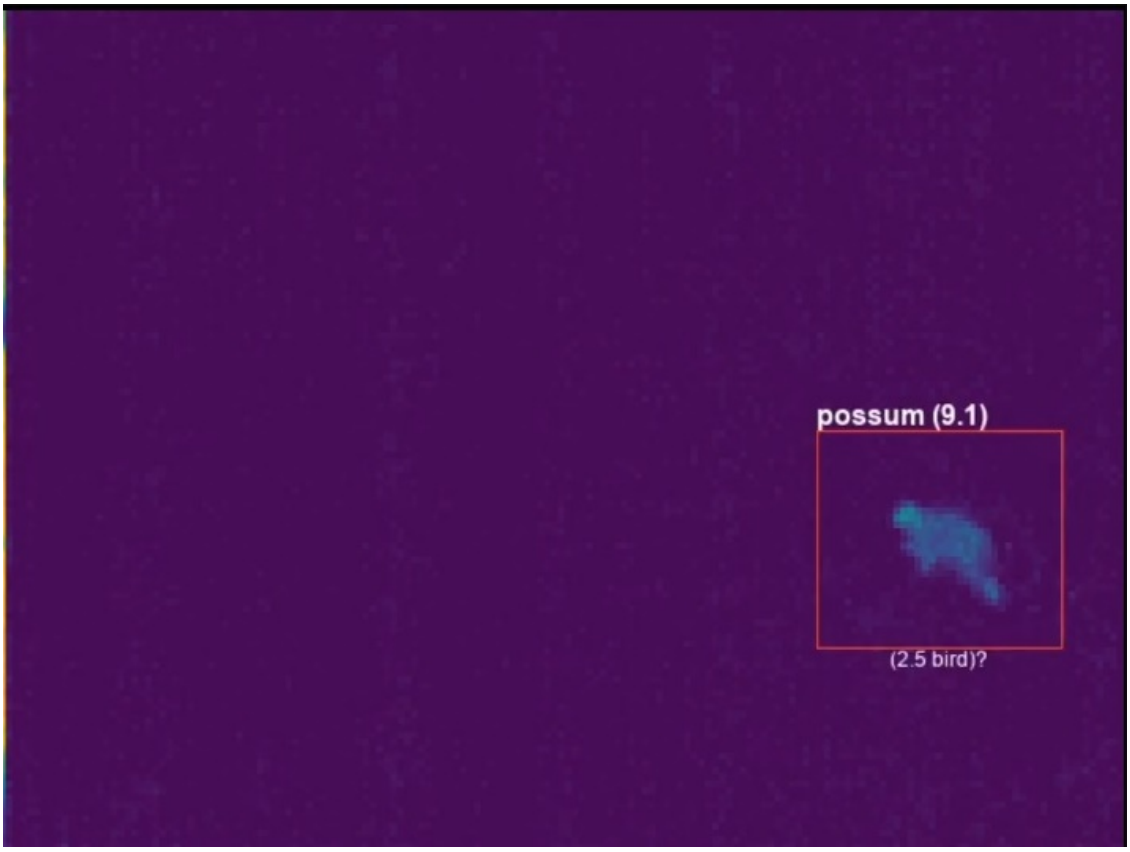


Figure 13: Frame of video after classifier has been run. The above number is the current best classification over the entire track (the peak certainty reached) while the number below is the current classification Cacophony.org website

for all classes False-positive scores are multiplied by 0.8 so that a strong false-positive identification is always beaten by a strong animal classification.

Track frames bumped up against the edge of the frame are also penalised.

The output of this entire process is a highlighted, classified, combined track of a single animal.

5.20 Evaluation

One last step in producing a model is to run the model on some CPTV files not seen during training. This will provide some insight into how well the model does on real data and will uncover any issues resulting from differences in the classification pre-processing as compared to the training.

The dataset also applies some filtering which will not be applied here, so the model will often perform worse on these clips.

The MPEG output from the evaluation step can give guidance on how well the model is performing on specific examples, and give a clear indication of which types of video setup give poor results (e.g. the camera being too close to the ground).

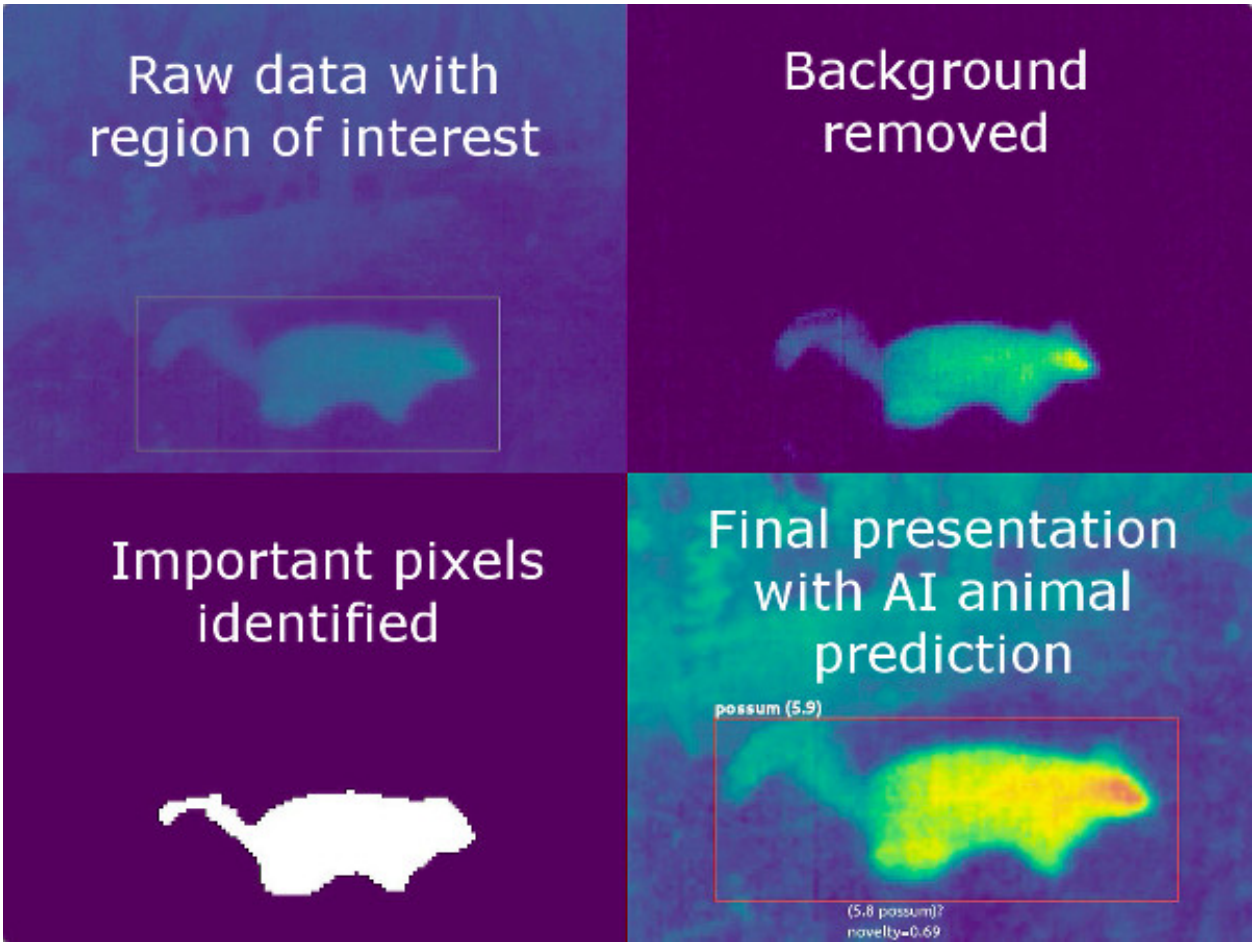


Figure 14: Machine Learning Steps

6 Results

6.1 ML Steps

Each step generates a visual output, shown in Figure 14.

6.2 Model Variations

Two variations of the model were designed;

A low quality "LQ" version, which is designed to run on the Raspberry Pi. This uses convolutional striding instead of a max pool. A max pooling process is where a group of pixels are downsampled via a filter, and assigned the maximum value found in the filter.

Convolutional striding is another process of downsampling, but the downsampled pixel is assigned whatever value is found at each stride length - e.g.,every 3rd pixel.

A higher quality "HQ" version is also provided which is designed to be run on a server, and offer the best possible classification accuracy. This typically takes 3x as long to train but gains a notable increase in accuracy. The differences are seen in Table 5.

Model	Model Training Time per Segment	Test Set Error
LQ	5ms	6.0
HQ	17ms	3.4

Table 5: Model Training Time

The models were also tested without including optical flow, which both reduces the model's computation and allows the very costly optical flow generation step to be skipped. Removing optically flow typically increases the error rate by about 5%. Although both models were designed and evaluated, the rest of the results are all based on the HQ model - as this is the "upper limit" of performance based on the data available.

label	**count**
bird	551
bird/kiwi	68
cat	288
dog	41
ferret	6
hedgehog	665
human	174
insect	147
mouse	18
other	7
possum	627
rabbit	80
rat	722
spider	2
stoat	446
weasel	5
unidentified	176
false-positive	813

Table 6: Thermal Recordings

6.3 Trap Deployment Locations

Name	Latitude	Longitude	Animal Visits
ospri15	-43.65315	172.63575	15
Hubble 1	-43.81065	172.97055	16
livingsprings05	-43.64775	172.63665	16
Awaawaroa BTG	-36.82485	175.10985	20
ospri14	-43.65135	172.63125	21
ospri20	-39.13425	173.96415	21
ospri18	-43.65315	172.63665	26
ashgrove1	-43.56045	172.63575	27
davidblake02	-36.94365	174.66165	32
ospri12	-43.65495	172.63215	42
ospri16	-39.05505	174.10455	47
Eliminator1	-43.81065	172.97055	48
ospri11	-43.65585	172.63125	63
ospri13	-43.65405	172.63485	65
Pourewa camera	-36.85815	174.81105	66
davidblake03	36.84825	174.76155	77
A_S4_C1	-43.65315	172.63215	84
A_S1_C3	-43.65315	172.63665	99
Hubble 2	-43.81065	172.97055	104
A_S4_C2	-43.65405	172.62945	108
A_S3_C2	-43.65585	172.62945	145
A_S1_C2	-43.65225	172.63845	155
A_S1_C1	-43.65045	172.63935	163
ospri17	-39.05505	174.10455	167
A_S3_C1	-43.65315	172.63215	168
A_S4_C3	-43.65315	172.62765	170
A_S2_C2	-43.65675	172.63125	171
ruru19w44a	-36.03915	174.51675	197
A_S3_C3	-43.65495	172.62855	218
A_S2_C1	-43.65405	172.63305	252
A_S2_C3	-43.65675	172.63125	269
TrapCam02	-43.65585	172.63125	562
TrapCam01	-43.65495	172.63125	658
Wallaby2	-44.76285	170.56395	905
TrapCam03	-43.65585	172.63125	953

Table 7: Unique Trap Locations (N.B. 8 locations with fewer than 10 visits (totalling 40 visits) have been omitted for clarity of the table.

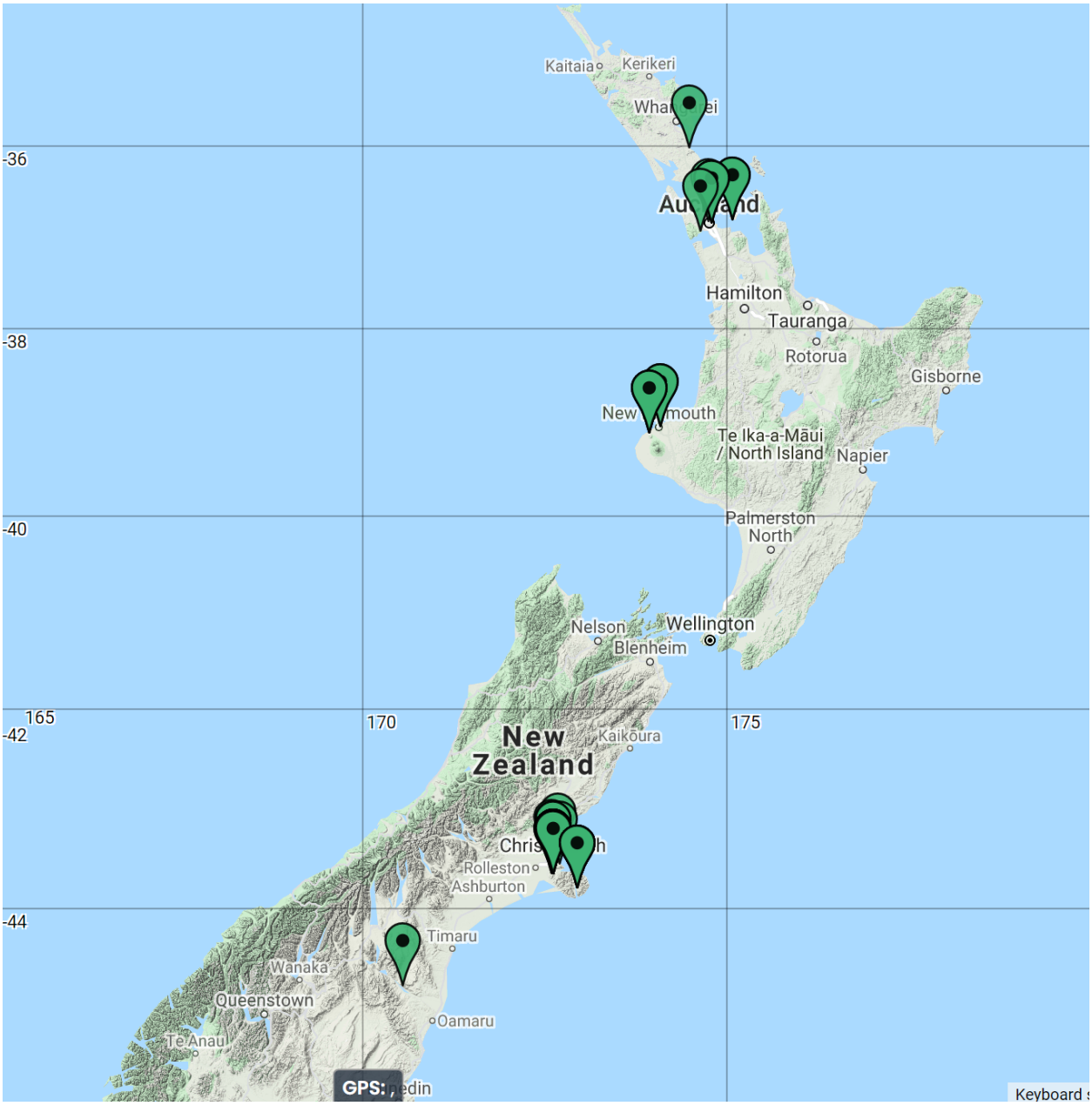


Figure 15: Map of camera locations

The traps were placed wherever volunteers were willing to monitor them, as such they are heavily focused around near-to-population patches of bush, as opposed to far into NZ wilderness. This was simply a limitation of labour-hours needed. Exact locations can be seen in Figure 15 and subsection 6.5.

6.4 Animal Visit Duration

Animal	Time Total(s)	Number of Recordings	Percent of Total Footage	Average Visit Duration(s)
bird	40963	1044	12.95%	39.2
cat	11965	185	3.78%	64.7
dog	1645	33	0.52%	49.8
false-positive	51412	2255	16.26%	22.8
hedgehog	25506	518	8.07%	49.2
human	2022	59	0.64%	34.3
insect	887	45	0.28%	19.7
leporidae	3532	104	1.12%	34.0
mustelid	458	9	0.14%	50.9
other	20	1	0.01%	20.0
part	2847	81	0.90%	35.1
poor tracking	135	6	0.04%	22.5
possum	32190	561	10.18%	57.4
rodent	76064	1889	24.05%	40.3
sheep	3008	59	0.95%	51.0
unidentified	26959	1235	8.53%	21.8
wallaby	36616	1890	11.58%	19.4

Table 8: Average Animal Footage duration, (trained classes highlighted)

6.5 Data Collection Spread

The data collected are widely spread and there is very little consistency between discrete data sources. As shown in ; different cameras detect wildly different total numbers of recordings, different ratios of species, and different hours of activity. Similarly, different individual animals spend much different amounts of time on screen, with very little consistency both within species, and between them as shown in Table 8. Each recording is totally discrete from the others. This, of course, is as expected for cameras arrayed across New Zealand. However, it does imply there there is no systemic error with regard to animal detection (e.g. if there was no detection of possums anywhere, or an exceedingly high detection of rabbits everywhere, it would imply that the recordings are not a true representation.

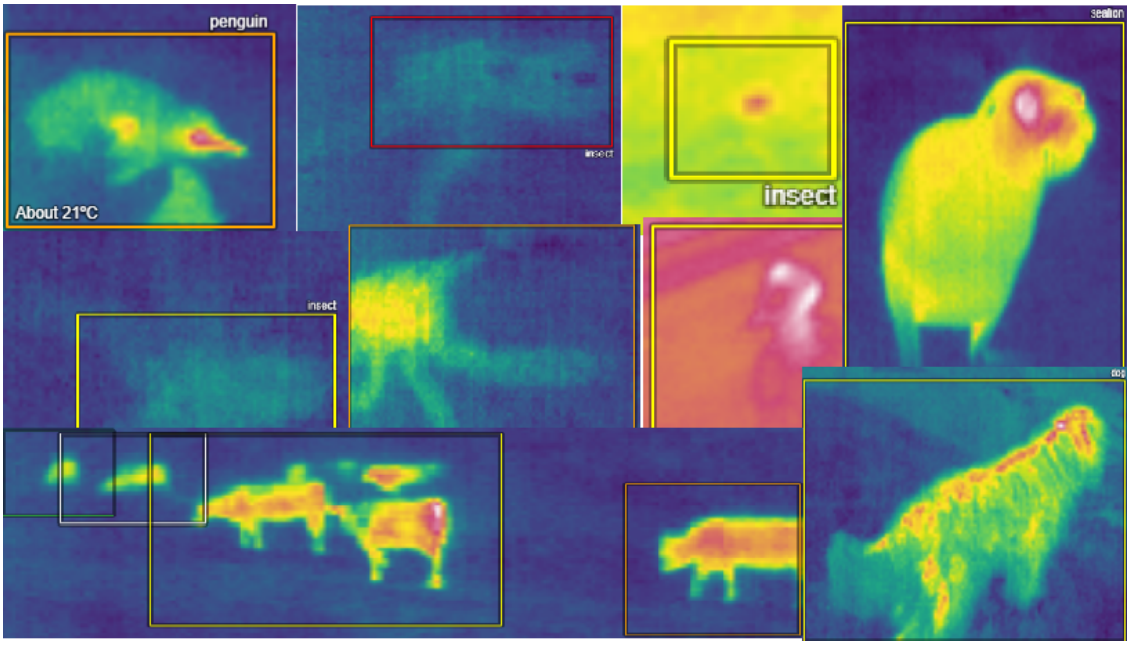


Figure 16: Untrained Animal Class Footage

6.6 Animal Labelling Accuracy

	Accuracy All Classes	Accuracy excluding false positives
Number of correct Tags	4439	3259
Total Tracks	9974	5475
Tag Accuracy ratio	0.4450571486	0.5952511416

Table 9: The overall tag accuracy

The overall accuracy of the Net, which is defined here as simply the number of correct tags over the number of tracks shown to the net, is unsurprisingly low, at 44.5% as shown in Table 9. This includes classes that the net has not been trained on, as well as false positives (footage shown to the net which does not include any animal at all). Although the net could be trained to exclude false positives on its own, that was determined to be outside the scope, and as such, the false positives should be detected by the pre-processing operation. When these false positives are excluded, the accuracy increases to almost 59.5%, also shown in Table 9.

This is similarly low as it includes a number of tracks being shown to the net that it will never be able to identify, as it has simply not been trained on them, these animals include dogs, humans, insects, sheep, sealions, penguins, and wallabies. Though not represented here, there are some as well which are not supported even by human tagging on the website - including deer, tahr, and reptiles. The net, however, does support "unidentified" animals - which is a class that means that even humans cannot identify them; which includes situations such as the animal moving at high speeds. being too close or too far from the camera, or an a-typical example that is so strange that even humans lose the ability to identify them, such as an animal that has been disfigured or dismembered (e.g. a possum without a tail). The quality of the visit is usually high enough that a classification *could* be made, if the net was trained and had enough data to recognize the class, as shown in Figure 16. All these edge cases (or future cases) contribute to the low overall accuracy.

However, when excluded, as shown in Table 10, the accuracy is much higher. When the net is only graded on its ability to correctly tag trained classes, the net reaches an average accuracy of 87%, which its best class being the highest-data volume class of rodents at 95% accuracy, and its lowest class being its lowest data-volume class of cats, at only 62% accuracy.

Correct Tag Per Trained Animal Class	CRNN Tagged	Human Tagged(total)	Accuracy Ratio
bird	683	758	0.90
cat	79	128	0.62
hedgehog	309	392	0.79
leporidae	79	82	0.96
possum	341	383	0.89
rodent	1045	1103	0.95
unidentified	304	320	0.95
Total	2840	3166	0.87

Table 10: AI tagging accuracy only including trained classes

6.7 Analysis

The average visit duration, though interesting, does not impact the neural net, as it is able to classify animals with as little as a 3 second footage time. However, the average visit duration for all animals is over 19 seconds, and while there has not been any behavioural analysis performed in this project, the primary reason is the field of view of the camera captures animals for a long time, and animals in the frame are generally not "rushing" anywhere - the usual behaviour appears to be foraging, with the animals moving slowly.

The accuracy of the neural network when counting only classes it had been trained on was roughly 86.5 percent. However when cats are excluded, this accuracy increases to 90 percent. This is likely due to limited available footage of cats to train the CRNN on, which has predictably led to a low classification accuracy. Furthermore, the CRNN consistently missclassifies cats as possums, due to the overall similarity of body-shape, and the extreme volume of footage of possums. As such, although there is a large data availability of possums, their accuracy is slightly lower than could be expected, due to the false positives of classifying cats as possums.

Of high interest, however, is that cats - while part of the training set - have relatively little observations, while wallabies - not part of the training set - have plenty. this can be used to inform future development into the value of adding wallabies as a trained class. Wallabies are pests, though not nearly as damaging to the bird life as cats.

Figure 17: Living Springs and Christchurch

6.8 Trapping method performance in Living Springs

Other than habitat loss, the main threat for natives on Banks Peninsula are the invasive mammals. Currently 15 introduced species roam free including rodents, mustelids, rabbits and possums [15]. Predator Free NZ is split into region, with the branch 'Pest Free Banks Peninsula (PFBP)' set-up in 2018 by a collaborative of 14 founding local organisations. The programme focuses on widespread predator control to preserve endemic biodiversity. PFBP have funded field-testing for the Cacophony Project, which takes place on private land owned by Living Springs as shown in Figure 17. Living Springs is a camp site and conference centre located in the Port Hills at the head of the Allandale Valley on the western perimeter of Banks Peninsula. This privately owned land stems 420ha with three major gullies descending west to east. The highest elevations of 450 masl drops to 30 masl at the lowest point. The landscape is a mixture of native bush and pastureland. A long-term goal is to create contiguous bush linking to surrounding land, owned by private stakeholders and the council. The location is good for field-testing due to its abundance of invasive and native species. For monitoring the field site is split into three sections (A,B and C) using the gullies as natural

divides as shown in Figure 18

Figure 18: Living Springs Testing Sites and Sections

Current DOC Efforts

The Department of Conservation (DOC) is responsible for the majority of New Zealand's monitoring as establishing good, standardised practise allows clear overviews of the ecological health of the country. Monitoring exists at three levels: 1) Broadscale, 2) Nationally Managed Places and 3) Research (Localscale). All three segments provide a framework to assess performance and guide policy making. Regarding pest control, both biodiversity inventories and long-term monitoring programs need implementing[19]. As monitoring uses parameters at predetermined frequencies to measure trends in populations, it is a key element for predator control programmes as the success of interventions can be quantified. Several methods have been established to optimise monitoring across the wide range of pest species in New Zealand.

	Tracking Tunnels	Chew Cards	Waxtags	Catch	Faecal Pellets	Dist Samples	Night Counts	CPUE Indices
Rodents	X	X						
Mustelids	X	X						
Possums		X	X	X				
Deer					X			
Wallabies						X		
Rabbits							X	
Goats								X

Table 11: Trapping Tools Deployed by Doc in Living Springs

For smaller pests, excluding rabbits, the most common techniques are tracking tunnels and chew cards. These are cheap, easy to use, and target multiple species thus are considered efficient to monitor pest distribution [19]. Tracking tunnels are principally used for rodent and mustelid detection. They consist of corflute plastic folded into a tunnel with middle section covered with tracking ink. To entice animals' tunnels are baited with food (e.g. peanut butter). Tracking tunnels are placed along transects spaced 50 m apart and deployed for up to a week. The resulting footprints can be identified and tracking rates calculated [19]. Chew cards are another commonly used monitoring method. These channelled cards are filled with scented baits such as peanut butter, aniseed paste or soft meats. They are then attached to trees or posts for up to a week. The resulting bitemarks can be analysed using guides to identify rodents, mustelids or possums. Chew cards have higher detection rates than tracking tunnels and are relatively inexpensive therefore are often used for large-scale monitoring [19]. The techniques and which animals they were used to record at Living Springs can be seen in Table 11

Despite their wide usage, these methods are extremely labour intensive and limited in data output. Therefore, to achieve pest eradication by 2050 other methods are needed to optimise wide-scale monitoring with reduced labour costs. Trail cameras are increasingly used globally due to their ability to study mammal occupancy, abundance, behaviour and distribution[51]. The advanced development of infra-red trail cameras has greatly improved data output, with high quality photographs/videos stored onto memory cards and able to work nocturnally [36]. Despite their high cost, it has been evidenced that their use is cost-effective over long-term studies (e.g. 5 years). Long-term studies also enable higher data output, thus explaining the method's popularity for large-scale monitoring programmes [45]. Although conventional trail cameras have evolved significantly in sensitivity due to their high resolution, and are available for relatively cheap prices, they still present some major flaws. Trail cameras assume that detection is constant, however imperfect detectability is a common sampling error [1]. This has been highlighted in a study in North Carolina where

the Passive Infrared Motion (PIR) detection in a model of trail camera often did not trigger and missed up to 14–16% of events with large, identifiable mammals[70]. Smaller species also cause a challenge as identification is difficult unless they remain still, unobstructed by vegetation and at close range. Several smaller mammals and birds often evade detection altogether as the PIR is not triggered [70].

Traditional Trapping and Surveying Techniques// Prior to this experiment traditional monitoring was undertaken from the 16th June-23rd June 2020. Six transects were chosen in Section A avoiding overlap with proposed locations for the thermal camera transects. Forty baited chew cards were deployed along four transects at 20 m spacing (10 cards per line). Twenty tracking tunnels were deployed along two transects for tracking tunnels spaced at 50 m (10 tunnels per line). Both were baited with peanut butter and left for 7 days before collection. The data was then uploaded into Trap NZ and the Chew Card Index, Tracking Tunnel Index and Predator Abundance Index calculated.

The study site was Living Springs and monitoring was limited to Section A (48 hectares) due to thermal camera availability and time constraints. The first half of the study was conducted across two periods: Winter (5th-26th August 2020) and Spring (2nd-30th September 2020), each survey lasting 4 weeks.

After each month, the datasets were downloaded as csv files, and manipulated to fit into Trap NZ. TrapNZ is an online service allowing monitoring records to be uploaded from multiple sources to store, present and analyse data. Presence/absence tables were made to calculate occupancy of each transect line per pest and nontarget species. Occupancy per species across the study site was calculated as the Predator Presence Index (PPI). In addition, a simple Visit Abundance Index (VAI) was calculated. The results that previous traditional monitoring yielded also had basic statistics calculated including the Chew Card Index (CCI), Tracking Tunnel Index (TTI), and Predator Abundance Index (PAI). The source of each quantification metric can be seen in Table 12

Indices		Method	Definition	Sources
Chew Card Index	CCI	Chew Cards(CTC)	Proportion of Cards Bitten	Ruffel Innes, Didham, 2014
Tracking Tunnel Index	TTI	Tracking Tunnels	Proportion of tunnels with tracks	Blackwell et al. 2002
Predator Abundance Index	PAI	Chew Cards * Tracking Tunnels	Proportion of relative abundance calculated for all lines in the study	DOC, 2012
Predator Presence Index	PPI	Cacophony Thermal Cameras	Proportion of Presence of Given Species cross the area	The Cacophony Project
Visit Abundance Index	VAI	Cacophony Thermal Cameras	Mean number of Visits to a device by a given speces	The Cacophony Project

Table 12: Quantification Indices

Thermal Camera detection rates compared with other trail cameras

Four cameras were set up, two close to a bait station, and two further away (for a wider field of view). Of the two at each site, one was the designed (Cacophany) thermal camera, and the other was a trail camera (Bushnell Essential E2).

	Thermal	Other Trail Cameras	Times Better
Detections close camera	50	4	13
Detections far camera	50	1	50
Video length close camera (seconds)	3074	28	110
Video length far camera (seconds)	3074	10	307

Table 13: Table comparing thermal camera detection rate with other trail cameras

A visit was determined by looking through all the videos and if there were multiple videos around the same time it counted as only one visit. For example, there may have been 5 videos all at the same time so that is counted as one visit.he thermal camera also saw 11 cats and 9 hedgehogs but only rats are of primary interest for this comparison.

The close trail camera triggered 4 times but only during one visit was a rat caught on the video. These cameras have a delay in start up to conserve power and it shows they are often too slow to wake up from low power mode.

When rats did go in front of the trail camera they were often missed because the trail camera either didn't trigger or was too slow at "waking up". This isnt really surprising, as the trail cameras are primarily designed to detect animals about the size of wild-boar and larger.

As shown in Table 13, the thermal camera was many times more effective at detecting rats than other trail camears (between 100 and 300 times as sensitive).

Living Springs Results

In total the two seasons yielded 2026 results with 978 across the winter period and 1048 during spring as shown in Figure 19. For analysis, several data were excluded including insects, sheep, dogs, humans and false positives. The remaining results gave a total of 975 visits in winter and 1041 in spring (including 'unidentified' data). The data shown includes

the main pest species identified (Rodents, Mustelids, Rabbits, Hedgehogs, Possums and Cats), non-target species (Birds) and data tagged as 'unidentified' (presence of a species but unidentifiable). The results show high overall species occupancy during both winter (88%) and spring (75%), with the graph highlighting rodents having the highest no. of visits in both seasons. During winter rabbits were the only species not recorded, whereas during spring no mustelids nor cats were recorded. Unidentified recordings were also displayed to highlight imperfect detectability within the methodology. by given species across all four transect lines in Section A. Rodents and possums were the most prevalent pests recorded during both seasons. Mustelids and cats were only present in winter, whereas rabbits were only recorded during spring. Unspecified shows the largest visible difference between seasons with 136 recorded in spring vs. 38 in winter.

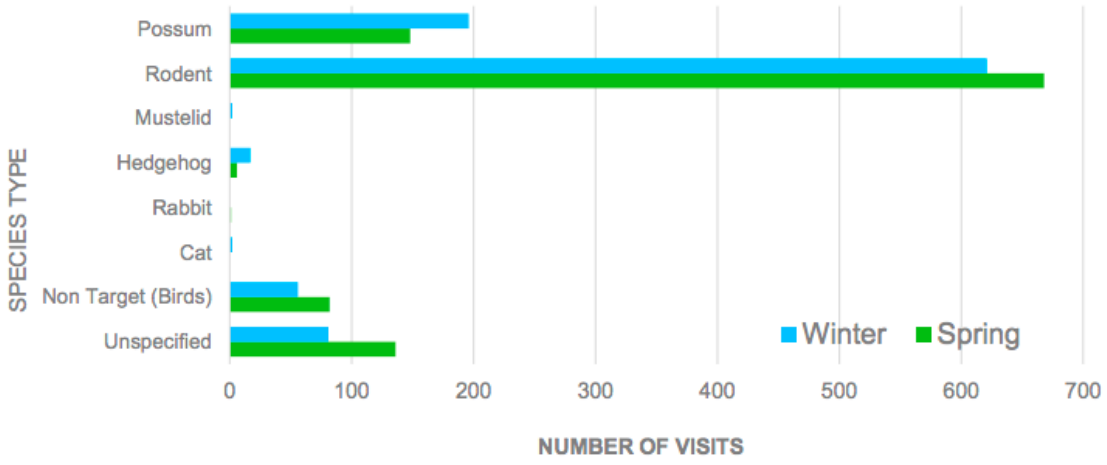


Figure 19: Total number of visits per season (Winter and Spring)

One objective for this comparative study was to determine data output quality and its use for analysis. The data gathered during this period is shown in Table 14. Occupancy data was the most valuable output at this stage. Occupancy is defined as the proportion of a total area where a target species is present [4]. For this study, the Predator Presence Index was used to calculate occupancy as the proportion of species present at any of the 12 monitoring stations. Areas of high occupancy were deemed hotspots and regarded as top priority for

	Total Visits		Mean VAI		Mean PPI	
Species	Winter	Spring	Winter	Spring	Winter	Spring
Possums	196	148	16.3	12.3	75%	67%
Rodents	621	668	51.6	55.7	100%	92%
mustelids	2	0	0.17	0	17%	0%
Hedgehogs	17	6	1.42	0.5	17%	17%
Rabbits	0	1	0	0.8	0%	8%
Cats	2	0	0.17	0	17%	0%
Birds	56	82	4.7	6.9	58%	75%

Table 14: The total visits and calculated VAI (Visit Abundance Index) and PPI (Predator Presence Index) values for each species during both winter and spring results. Unknown data was removed, but one non-target species (birds) was included.

pest control measures. As the goal of monitoring was to understand distribution of predators, using occupancy rather than relative abundance allows several limiting assumptions to be disregarded (e.g. pseudoreplication). For example, occupancy accounts for imperfect detectability by being robust to false absences (mackenzie2005issues). Therefore, absence of a species can be often attributed to factors other than detection probability.

Signs of mustelids are present at Living Springs, however only one video-capture occurred. In New Zealand stoats exhibit predominantly diurnal behaviour, thus making their detection probability extremely low in this study. However, ferrets and weasels display nocturnal behaviour and have been found in high abundance on Banks Peninsula, therefore it is unknown why the occupancy was low, therefore further data collection is needed [15]. Mustelids have high movement speeds and therefore some of the 'unknown' videos could have been ferrets or weasels, although this cannot be verified either. There is also the possibility sensitivity of the cameras needs to be increased to improve detectability of mustelids. A cat was also detected during this study, although whether feral or not could not be identified - however a cat located near native NZ bush is a a concern all same, whether feral or not. The thermal cameras could be used for targeting feral cats in areas where pets are known to be absent. At Living Springs, most birds are passerine species including the introduced common blackbird (Turdus merula), the native bellbird (Anthornis melanura) and the reintroduced native

NZ pigeon (Hemiphaga novaeseelandiae). Several birds were detected during this study but could not be identified (to 100% accuracy). However, several of New Zealand's endangered ground-dwelling birds which remain at highest threat from predators could be identified even at a high-level[53]. Using cacophony thermal cameras to monitor these species could highlight areas for protection, dispersal and establishment of translocated populations and assist in decision-making for pest control measures.

For more thorough analysis, occupancy of species could be calculated per transect for comparison of varying habitat type and elevation. Incorporating covariates into future field tests is important as time of night, home range, temperature, humidity, rainfall, and wind impact detection probability. Occupancy is also impacted by environmental factors (e.g. food sources) and disturbance (e.g. human activity) and is important to note if obtaining low occupancy results. Occupancy estimation provides more reliable results with repeated data. A larger sample size and incorporation of covariates would allow temporal and spatial population trends to be assessed. Within traditional monitoring DOC encourages using the CCI, TTI or PAI for direct comparison of habitat types, pre- and post-treatment studies or as a relative abundance estimate over time (DOC 2021). For the thermal cameras the Visit Abundance Index (VAI) did provide a basic statistic and often abudnance usually positively correlates with occupancy, however without the inclusion of covariates and noting that occupancy probability is resistant to abundance change, the current abundance results are not reliable. Although a relative abundance index in future studies could help understand population trends pre-and-post pest control measures and quantifying their success.

There were other important factors found about the data output. Data volume was high, yet limited initial labour was needed. The automation of uploads to the server provided an organised and fast way to access data. The ability to download data into a universal format (e.g. csv file) also provides an easy option for people to store and analyse their data, and not rely solely on Cacophony software. The ease of reduced labour yet high data volume results in a high scalability, which is needed if to be implemented nationally.

Device	Deployment Effort	Collection Effort	False Negatives	Data Quality	Data Volume	Scalability
Chew Card	Med	Med	High	Low	Low	Low
Tracking Tunnel	Med	Med	Med	High	High	Low
Trail Camera	Med	High	Low	V-High	V-High	Med
Cacophony Cam	Med	Low	V-Low	V-High	V-High	V-High

Table 15: Data Collection Tool Comparison(False Negatives defined as "not detected, when another tool in the same location successfully detected")

Multiple species detection was the other key advantage. The range of species was high, from tiny rodents to sheep. Additionally, all pest species known at Living Springs were detected, even if at a low occupancy (e.g. mustelids or rabbits). Not only were pests detected, but nontarget species (e.g. birds) were recorded. Traditional chew cards were used as lures, yet many recordings showed individuals never interacting with the lure. Therefore, highlighting the problem of imperfect detectability in traditional methods and evidencing improved interaction rate for the thermal cameras. Other factors contributing to improved interaction rate were identified as minimal habitat disturbance, lack of light source and high sensitivity to triggering. The Cacophony project has recently compared the main aspects of monitoring methodologies using data provided by DOC and Cacophony itself. A summary of the comparative advantages and disadvantages of methods is shown in Table 15.

Data Collection Tool Comparison

Despite having a lower resolution than advanced trail cameras, behavioural observation could still occur at a small scale. Individual behaviours (e.g. movement) and inter/intra-species interactions (e.g. predation) could be described. For example, Camera 2 on transect S4 captured a mustelid chasing a rodent during the winter period, however such an action cannot be reflected via other methods. Thus, showing there are additional uses for the thermal cameras than basic occupancy. Overall, this preliminary experiment showed high data quality and quantity without compromise to labour efforts, inferring its importance for predator control in New Zealand. One datapoint omission is "the cost of equipment" which is currently extremely high for thermal cameras, however these tools are entirely reusable

and currently early in their manufacturing life - still being made by hand. As such the life-time cost of these tools is going to change wildly in the next few years, making such a comparison at this time meaningless.

Living Springs CRNN field Test Accuracy

The second objective was to train the CRNN and determine its accuracy. The current version of the device already provides a more sensitive and automated way of analysing monitoring data than other monitoring methods. The CRNN is also more advanced than many other AI-integrated trail cams (often used by hunters, which simply highlight the presence of *any* animal). However, manual tagging was still needed as CRNN accuracy varied per species(and the training database is still growin in size). For example, in a Living Springs Cacophony Project dataset of 112 videos tagged as possums by humans, the CRNN had correctly tagged 69%. With more data there can be further CRNN training and improved accuracy. Despite needing manual tagging to ensure accurate results, the process was speedy and efficient. The ability to gather daily-timescale data with automated sorting of the videos with associated data (e.g.,date/time/GPS) and a user-friendly interface vastly reduced human labour. Future automation in tagging will also eliminate human bias once refined. Thus, the thermal cameras offer many of the advantages of trail cameras provide with the addition of reduced human labour. One downside of the current system is the inability to recognise multiple individuals of the same species in a video, however a tag can be added to train the CRNN further to refine this feature - essentially the more granular the human-tagging, the better the ability of the CRNN itself.

Comparing the Performance of the CRNN with traditional monitoring techniques

Each monitoring method yielded abundance/presence indices, however due to their differing value type, they cannot be directly compared statistically. Therefore, to visualise the differences in data volume/quality between thermal cameras and traditional monitoring, data were manually compared. A subset of the data duration was chosen due to the difficulty of

directly comparing this data - so a period of 7 days was chosen. Important differences found were that tracking tunnels can only identify rodents and mustelids but have the benefit of identifying diurnal species. However, thermal cameras can identify a wider range of species as well as providing behavioural observation. Whilst tracking tunnels could differentiate between rodents (mice and rats), and the thermal camera could not, this is often irrelevant in pest control measures as both are caught using the same trapping techniques. However, it is important to note if unable to distinguish then both rats and mice needed to be targeted due to their own interspecies relationships [15]. The largest difference between the methods was data volume, with the quantity of thermal camera data was 3333x higher than for the tracking tunnels over the same collection time. These magnitudes of difference in data volume are likely due to imperfect detection and interaction rate of the traditional methods. Further comparisons of traditional vs. thermal camera monitoring should be carried out in future research to evidence the validity of this result for use a key advantage over current methods used in New Zealand.

Living Springs Transect Lines

Shown in Figure 20; S1-4 are transect lines with 3 thermal cameras apiece, while lines A1-4 are chew cards, while lines A5-6 are Tracking tunnels. S1 = native contiguous bush (by stream) S2 = native contiguous bush, S3 = native patchy bush, S4 = forest edge. The difference in observed animals can be seen in Table 16.

Figure 20: Living Springs transect lines

	S1 Transect	A6 Transect
Tools	3 Cameras (5th-12th August)	10 Tracking Tunnels
Rodents	195	5 mice, 1 rat
Birds	1	0
Possums	2	0
Unspecified	2	0
Total	200	6

Table 16: Comparative table of Camera line and Tracking tunnel line

Limitations of CRNN camera trapping

Understanding limitations is a key part of any study and vital when developing surveying

methods. The main limitation of thermal cameras is their ability to only identify warm-blooded, nocturnal animals. Fortunately, almost all pest species in New Zealand fall into this category and were detected during the study[15]. Currently other organisations are using Cacophony equipment to monitor larger pests such as wallabies in other habitat types (e.g. open planes) and have highlighted limitations, such as during installation the camera must not encompass any sky as this affects the thermal imaging thus skewing the data, but can be difficult if needing a wider scope of view for larger animals - once again this is a only a minor problem in the NZ context, as the tree-canopy occludes the sky.

1. **High initial Costs**

   The Cacophony thermal cameras have a high cost (¿ NZ$300) due to their advanced technology - though a large portion of this cost comes from the fact that the devices are hand-built, bespoke solutions to a problem that currently has very little in the way of industry efficiency increases (due the the fact that thermal cameras are a not-widely used technology compared to infrared cameras). However limited devices are needed per site the cost thus can be considered an investment for longterm monitoring projects. Furthermore, the cameras do not require much maintenance or repair as there are no moving mechanical parts. It is also important to note the concept of Moore's Law, which theorises that the number of transistors on a chip doubles every two years. This is relevant as it outlines the ideology that every few years technology becomes more advanced, yet the relative cost is reduced. Therefore, the price of the thermal cameras will reduce as their technology is improved further evidencing their cost-effectiveness, similar to the trend with trail cameras. The numerous other advantages of the device could negate the cost especially for large funding bodies such as Pest Free Banks Peninsula - and when a certain goal has been achieved (such as pest free Banks Peninsula), then the cameras can be reasonably easily moved elsewhere.

2. **Closed Populations**

   In order to extrapolate to the population level, the population must be assumed to

be closed, or at least to have a constant rate of immigration/emigration - assuming that essentially all of the measurements being made are actually measuring one single population (however the definition of population is being made, whether "this forest" or "banks peninsula" or "New Zealand"). However, several statistical models (e.g., N-mixture) can account for this, thus can be incorporated into analysis when the study is complete [16].

3. **Pseudoreplication**

   Pseudoreplication is an abundant issue for monitoring and is defined as the erroneous treatment of non-independent data as independent (Jordan, 2018). Regarding video capture the assumption is that each individual per recording is different. As previously described, the Cacophony Project have reduced some pseudoreplication by creating a visit system in the software. Even with 100% accuracy in the CRNN and visits system, pseudoreplication can never be fully accounted for. For example, lures can alter behaviour increasing likelihood of returning to the monitoring location. Movement between camera location or transect is also possible due to species' home ranges or dispersal. The Cacophony project are using Living Springs as a field test site for a trap and hotspots can direct trapping experiments to target particular subsets of species (e.g. possums with joeys). The CRNN pipeline is some-what robust to pseudoreplication based on the combination of multiple occurrences of animals into a single "visit". However there can never be any guarantee that two different animals of the same species visit in a short timeframe, nor that a single individual visits the same site multiple times with an hour between each.

4. **Individual level Identification**

   The low resolution of the thermals cameras reduces manufacturing costs, increases device robustness and battery life. However, it also makes the identification of animals at a individual level impossible, by the CRNN *or* by a human expert - there is simply

not enough data due to the lack of resolution and colour.

5. **Family Level Identification**

   Stoats, weasels and ferrets are morphological different, yet are often indistinguishable with the thermal cameras and can only be classed at family-level (Mustelidae). It is also difficult to differentiate rodents; rats and mice form part of the superfamily Muroidea but encompass multiple species. This limitation affects in-depth analysis as only generic trends at family-level can be done. However, presence of any of these species still indicates a need for pest control and measures usually target all members at a family level.

6. **Mechanical Challenges**

   There are also some mechanical differences that, while not exclusive to the camera measurement method, are still challenges faced by any monitoring techniques. Spot-checking data should occur even when the CRNN is highly accurate, due to issues such as animals knocking over the camera. However, in areas with signal the traps may be remotely monitored for failure.

   Areas of no signal provide the biggest challenge due to needing manual collection. Though labour time is much lower than with traditional trapping techniques, it still prevents truly real-time monitoring, a key advantage for the user.

7. **Unidentified Data**

   Despite the thermal cameras' high sensitivity and capture rates, there are still issues with unidentified data. Approximately 11% (8% in Winter and 13% in Spring) of the dataset were classed as unknown in the Living Springs Field Study, with 41% of the total footage seen not being able to be labelled by the AI. This was often due to obstruction by vegetation, untrained Data, direct blocking of the camera, or being partially in view. These data were excluded from statistics calculated but were kept in the overall results to highlight this as a limitation and to be accounted for in analysis

of the results. The AI's performance of 87% when only analysing "optimal data"10 still means for 13% wrong data, of simply miss-labelled data.

7 Conclusion

This paper proposes a new approach to species surveying, utilising a CRNN. By using breakthroughs in neural network architectures and designs, as well as modern hardware, new approaches are now possible that have not yet been investigated. Analysing thousands of hours of footage is now possible, and allows for more accurate, timely, and interesting surveying footage, far surpassing current approaches used by conservation programs. Prior to this research, a reliable dataset of thermal images did not exist, much less a dataset that records motion. Further, the data has been labelled, and categorised by location and time. While the creation of this dataset alone is a contribution, the implemented CRNN has a high performance and reliable detection for all trained classes, which increases as more data is gathered. This puts this neural network approach ahead of any other existing method, as those that do exist either use static images, infrared illumination, or perform only during the day.

This implementation is better at detecting animals than current low tech trap or observation based approaches by over 3 thousand times. Further, it is more accurate than extant trail cameras for detecting small mammals - being about 10-50 times better in experimental trials. The catchment area also increases linearly with more cameras (and thus) more footage, however has a much lower likelihood of "trap exhaustion" whereby pests learn quickly to avoid trapped areas if the trap density is too high.

Furthermore the network itself performs well on trained classes, with the accuracy of the CRNN reaching up to 87 percent accuracy and the catchment includes all night hours (the period of which can be increased or decreased based on latitude and time of year, or simply ambient light levels) - the filming technique uses an FLIR camera, and requires a cold background. Processing time (per occurrence) is unaffected by total footage (3ms processing time per animal-occurrence), though obviously the more footage captured, the more that needs to be processed, also increasing linearly.

8 Future Work

Three main areas where for future research focus are improving the three stages in the main processing pipeline; changes to the data and input, changes to the architecture and specifics of the network itself, and changes to the output/classification system. Furthermore if such improvements are successful, a more rigorous criteria by which to evaluate the system will be important. These criteria will include metrics such as accuracy, training time, classification time, data/input required to make a classification, sensitivity to novelty, and memory required.

The study so far shows there is the potential for the Cacophony thermal cameras to provide an efficient monitoring method with high scalability and reliability with a cost-beneficial investment and low labour requirements. The equipment could then be used nation-wide to make fast yet informed decisions for pest control measures. Further research relating to the thermal cameras is also in motion, investigating the sensitivity of the cameras and additional technologies (e.g. audio lures). This experiment has also helped shape the protocol for the thermal cameras. This protocol can then be implemented as a standardised method in NZ alongside traditional methods. Cacophony thermal cameras and the set deployment methodology is already being accepted at a regional level as an integral part of the Pest Free Banks Peninsula project and will likely be adopted by multiple local organisations within a couple of years.

8.1 Data and Input

Experiments run in this stage will change the way that the data is presented to the machine, which also means that they change each of the follow up steps - changing the way the data flows through the network.

- Track based input instead of clip based
  - Currently classification based on tracks (an individual animal, over a short period

of time), however the human labelling is performed on entire clips (up to 10 minutes of footage). An increased granularity of human labelled data would provide more training data, and reduce the amount of data that is currently discarded (for example, if two different animals appear in the same 10 minute clip, there is no way to label which one is which).

- Segment lengths
  - Currently all segments are 3 seconds long, or 27 frames. This length is semi-arbitrary, and could be extended or shortened. The LSTM units look back 40 steps, so this length could be extended somewhat. However the longer the segment is, the more frames that must be processed and therefore the higher the cost for classification. This is an optimization question of essentially "how short can the segment be, to still get the best results", as well as determining the importance of accuracy vs cost.
  - Furthermore, adding a sliding window/confidence threshold could allow the segment to be of a variable length, allowing the network to "be as long as needed to make a classification of a certain confidence, and no longer". Which in turn would allow the model to process certain "easy" classifications with minimum costs, and complicated classifications with higher confidence, instead of opting for a segment length that attempts to achieve both.

8.2 Network Architecture

This is the primary avenue for experimentation. Changing the fundamental architecture of the network may have the most significant impacts on the way the network will perform, and deciding and designing the "perfect" architecture may be challenging, as it is likely there will always be iterative improvements to be performed. As such, the experiments here may result in extreme changes to the architecture. While there are smaller, internal changes that

may be performed - e.g.,changing learning rate or adding a single extra layer to the network, it is unfeasible to test every variation.

- Using a pretrained net (likely trained on Cifar 100 (32*32 images), or Tiny Imagenet(64*64), which more closely match the size of the data input frames (48*48px)
  - A pretrained network may give better results, particularly if it is pretrained on similarly sized images. There are advantages to using pretrained networks even if the tasks are substantially different (such as a network pretrained on daylight images being used for thermal video frames). This is because even though the higher level weights may essentially be treated as random initialization, the lower levels which look for small characteristics such as edges and corners are still valuable.
- Use many more layers (e.g.,Resnet or Inception)
  - This is again an optimization problem. Having more layers has an increased processing cost as well as more memory, training, and processing costs. However, the accuracy is also increased, and trialling the use of more layers or networks which better facilitate additional layers will show whether this trade off is worthwhile.
- Comparing RNN against purely convolutional models such as 1D CNN instead of RNN, or a 3D CNN
  - A 1D or 3D CNN would be a simpler architecture than an RNN. The use of a purely convolutional model may give interesting results, and the simplicity may allow for the model to be more easily tweaked and finely tuned for my specific application.
  - A 1D CNN where there is essentially one net that operates on the image, and one net that operates on the optical flow, and then both feed into a time series CNN.
  - A 3D CNN would be able to take an input and perform a convolution that includes the change in time (multiple frames), which would allow for the network to make

motion based classifications, and therefore optical flow may not be needed.

  - A possible complication here is that the use of optical flow and an RNN allows for sequential feeding of arbitrary numbers of frames, where the RNN is always processing the most recent frame, using the previous frame to inform itself about motion. As this application is ideally using real time motion and tracking, this may be the ideal approach.

- Optical Flow

  - Optical flow is currently used to add extra information about the way an object moves within the frame. The additional information helps the network by giving information about the past. However, if the convolutions overlapped in time, then knowledge of motion would be inherently included in the network and would not need to be specifically added via an optical flow algorithm.

8.3 Output and Classification

Experimentation with the output would essentially be shifting the goalposts, and redefining success for the network. This may help the network be trained more quickly, or may make classification more difficult, depending on the experiment.

- Hierarchical classification

  - The use of hierarchical classification allows for more granularity in labelling of species, as well as grouping of common features to help make high level decisions more quickly.

  - Hierarchical classification is where an animal is labelled first into a class, and then into a subclass: "Mustelidae, then Stoat" or "Bird, then Kiwi".

  - In general, hierarchical classification is helpful when there are thousands of classes, and is used to diminish the realm of possibilities more quickly than using a single

classifier to decide on all classes, additionally it can help with degrees of correctness, whereby the network may have the family or class correct, but the subclass incorrect.

  - hierarchical classification can also help when the degrees of correctness are more easily obtained at different levels. For example, it may be very easy to determine if an image is of a car vs a building, but much more difficult to determine if it is a hospital or bar. One solution is to simply limit the number of classes and simply stop classification when the network decides it is a building, but this may lose valuable information, and in particular to this project, it is important to note what species a specific bird or pest is.
  - One large complication of hierarchical classification is that it can be challenging to balance the dataset at both the class and subclass level. Again this is of particular note for this project as the data is being collected as the model is being designed, and so I do not have tens of thousands of images to train on.

- Custom confusion cost

  - A confusion cost essentially adds a bias to make a decision one way or another, or can be thought of a "resistance" to making a certain classification. While many classification networks are designed to simply reach the highest level of accuracy, and any wrong answer is bad, in this project some "wrong answers" are worse than others. The obvious examples being misidentifying birds as pests, or humans as pests. it is very important that *no birds are identified as pests*, and the cost of identifying some pests as birds is much lower.
  - As such, training the network to make a classification based on: "if it is unclear if this is a bird or a pest, err on the side of classifying as bird" is important, and adding a customized (and non-symmetrical) confusion costs will help tailor the network's output.

References

[1] Kevork Abazajian et al. "The second data release of the sloan digital sky survey". In: *The Astronomical Journal* 128.1 (2004), p. 502.

[2] Igor Aizenberg, Naum N Aizenberg, and Joos PL Vandewalle. *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Springer Science & Business Media, 2013.

[3] Jimmy Ba and Brendan Frey. "Adaptive dropout for training deep neural networks". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3084–3092.

[4] Larissa L Bailey, Theodore R Simons, and Kenneth H Pollock. "Estimating site occupancy and species detection probability parameters for terrestrial salamanders". In: *Ecological Applications* 14.3 (2004), pp. 692–702.

[5] Pierre Baldi and Peter Sadowski. "The dropout learning algorithm". In: *Artificial intelligence* 210 (2014), pp. 78–122.

[6] Joseph Banks. *Joseph Banks's Journal*. Provided by New Zealand Department of Conservation, 1770.

[7] Dana M Bergstrom et al. "Indirect effects of invasive species removal devastate World Heritage Island". In: *Journal of Applied Ecology* 46.1 (2009), pp. 73–81.

[8] Michael Bode, Christopher M Baker, and Michaela Plein. "Eradicating down the food chain: optimal multispecies eradication schedules for a commonly encountered invaded island ecosystem". In: *Journal of Applied Ecology* 52.3 (2015), pp. 571–579.

[9] Cacophony.org website. *Cacophony Website*.

[10] B Kay Clapperton, SM Phillipson, and AD Woolhouse. "Field trials of slow-release synthetic lures for stoats (Mustela erminea) and ferrets (M. furo)". In: *New Zealand Journal of Zoology* 21.3 (1994), pp. 279–284.

[11] B. Clapperton. "A Review of the Current Knowledge of Rodent Behaviour in Relation to Control Devices". In: *Science for Conservation* 263 (Mar. 2006).

[12] *COCO dataset*. URL: `http://cocodataset.org` (visited on 08/06/2018).

[13] Franck Courchamp, Rosie Woodroffe, and Gary Roemer. "Removing protected populations to save endangered species". In: *Science* 302.5650 (2003), pp. 1532–1532.

[14] PE Cowan. "The influence of lures and relative opportunity for capture on catches of brushtail possums, Trichosurus vulpecula". In: *New Zealand journal of zoology* 14.2 (1987), pp. 149–161.

[15] M. Curnow and G.N. Kerr. *Predator Free Banks Peninsula: Scoping Analysis*. Land Environment and People research report. Lincoln University, 2017. ISBN: 9780864764102. URL: `https://books.google.co.nz/books?id=5JcBzgEACAAJ`.

[16] Francisco V Denes, Luis Fabio Silveira, and Steven R Beissinger. "Estimating abundance of unmarked animal populations: accounting for imperfect detection and other sources of zero inflation". In: *Methods in Ecology and Evolution* 6.5 (2015), pp. 543–556.

[17] New Zealand Department Of Conservation. *Department Of Conservation*. URL: `https://www.doc.govt.nz/our-work/monitoring-and-reporting-system/` (visited on 08/06/2018).

[18] New Zealand Department Of Conservation. *Department Of Conservation*. URL: `https://www.doc.govt.nz/our-work/biodiversity-inventory-and-monitoring/animal-pests/` (visited on 08/06/2018).

[19] Department Of Conservation, New Zealand. URL: `https://www.doc.govt.nz/Documents/our-work/predator-free-2050.pdf` (visited on 08/17/2018).

[20] Tim S Doherty et al. "Invasive predators and global biodiversity loss". In: *Proceedings of the National Academy of Sciences* 113.40 (2016), pp. 11261–11265.

[21] Lourdes Esteva and Hyun Mo Yang. "Mathematical model to assess the control of Aedes aegypti mosquitoes by the sterile insect technique". In: *Mathematical biosciences* 198.2 (2005), pp. 132–147.

[22] Kevin M Esvelt et al. "Emerging technology: concerning RNA-guided gene drives for the alteration of wild populations". In: *elife* 3 (2014), e03401.

[23] *FLIR dataset*. URL: `https://www.flir.com/oem/adas/adas-dataset-form/` (visited on 08/06/2018).

[24] Neil J Gemmell et al. "The Trojan female technique: a novel, effective and humane approach for pest population control". In: *Proceedings of the Royal Society B: Biological Sciences* 280.1773 (2013), p. 20132549.

[25] Juan B Gutierrez and John L Teem. "A model describing the effect of sex-reversed YY fish in an established wild population: the use of a Trojan Y chromosome to cause extinction of an introduced exotic species". In: *Journal of Theoretical Biology* 241.2 (2006), pp. 333–341.

[26] Richard HR Hahnloser et al. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit". In: *Nature* 405.6789 (2000), p. 947.

[27] William D Hamilton. "Extraordinary sex ratios". In: *Science* 156.3774 (1967), pp. 477–488.

[28] Tim Harvey-Samuel, Thomas Ant, and Luke Alphey. "Towards the genetic control of invasive species". In: *Biological Invasions* 19.6 (2017), pp. 1683–1703.

[29] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[30] Álvaro Huerta Herraiz, Alberto Pliego Marugán, and Fausto Pedro García Márquez. "Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure". In: *Renewable Energy* 153 (2020), pp. 334–348.

ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2020.01.148. URL: https://www.sciencedirect.com/science/article/pii/S0960148120301701.

[31] Vijay John et al. "Pedestrian detection in thermal images using adaptive fuzzy C-means clustering and convolutional neural networks". In: *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. 2015, pp. 246–249. DOI: 10.1109/MVA.2015.7153177.

[32] Kate Gudsell, New Zealand Radio, New Zealand. *Four out of five NZ bird species in trouble*. URL: https://www.doc.govt.nz/our-work/monitoring-and-reporting-system/ (visited on 08/17/2018).

[33] EF Knipling. "Possibilities of insect control or eradication through the use of sexually sterile males". In: *Journal of Economic Entomology* 48.4 (1955), pp. 459–462.

[34] R Keller Kopf et al. "Confronting the risks of large-scale invasive species control". In: *Nature Ecology & Evolution* 1.6 (2017), pp. 1–4.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[36] Thomas E Kucera and Reginald H Barrett. "A history of camera trapping". In: *Camera traps in animal ecology*. Springer, 2011, pp. 9–26.

[37] Adam Lampert et al. "Optimal approaches for balancing invasive species eradication and endangered species management". In: *Science* 344.6187 (2014), pp. 1028–1031.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[39] Adam Ligocki et al. "Fully Automated DCNN-Based Thermal Images Annotation Using Neural Network Pretrained on RGB Data". In: *Sensors* 21.4 (2021). ISSN: 1424-8220. DOI: 10.3390/s21041552. URL: https://www.mdpi.com/1424-8220/21/4/1552.

[40] Cai Lile and Li Yiqun. "Anomaly detection in thermal images using deep neural networks". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 2299–2303. DOI: 10.1109/ICIP.2017.8296692.

[41] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[42] Kenneth A McColl, Agus Sunarto, and Matthew J Neave. "Biocontrol of carp: more than just a Herpesvirus". In: *Frontiers in microbiology* 9 (2018), p. 2288.

[43] Paul D Meek et al. "Camera traps can be heard and seen by animals". In: *PloS one* 9.10 (2014), e110832.

[44] Engineering National Academies of Sciences, Medicine, et al. *Gene drives on the horizon: advancing science, navigating uncertainty, and aligning research with public values*. National Academies Press, 2016.

[45] Allan F O'Connell, James D Nichols, and K Ullas Karanth. *Camera traps in animal ecology: methods and analyses*. Springer Science & Business Media, 2010.

[46] Dean R Paini et al. "Global threat to agriculture from invasive species". In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7575–7579.

[47] Maulik R Patel et al. "A mitochondrial DNA hypomorph of cytochrome oxidase specifically impairs male fertility in Drosophila melanogaster". In: *Elife* 5 (2016), e16923.

[48] Cacophany Project. *Cacophany Project Motion Detector Tuning*. URL: https://cacophony.org.nz/tuning-motion-detector (visited on 06/22/2021).

[49] Prajit Ramachandran, Barret Zoph, and Quoc V Le. "Searching for activation functions". In: (2018).

[50] Hugh Alexander Robertson et al. *Conservation status of New Zealand birds, 2012*. Publishing Team, Department of Conservation, 2013.

[51] Francesco Rovero et al. “” Which camera trap type and how many do I need?” A review of camera features and study designs for a range of wildlife research applications.” In: *Hystrix* 24.2 (2013).

[52] James C Russell and Keith G Broome. “Fifty years of rodent eradications in New Zealand: another decade of advances”. In: *New Zealand Journal of Ecology* 40.2 (2016), pp. 197–204.

[53] James C Russell et al. “Predator-free New Zealand: conservation country”. In: *BioScience* 65.5 (2015), pp. 520–525.

[54] Dominik Scherer, Andreas Müller, and Sven Behnke. “Evaluation of pooling operations in convolutional architectures for object recognition”. In: *International conference on artificial neural networks*. Springer. 2010, pp. 92–101.

[55] Martin A Schlaepfer, Dov F Sax, and Julian D Olden. “The potential conservation value of non-native species”. In: *Conservation Biology* 25.3 (2011), pp. 428–437.

[56] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural networks* 61 (2015), pp. 85–117.

[57] Haidong Shao et al. “Intelligent Fault Diagnosis of Rotor-Bearing System Under Varying Working Conditions With Modified Transfer Convolutional Neural Network and Thermal Images”. In: *IEEE Transactions on Industrial Informatics* 17.5 (2021), pp. 3488–3496. DOI: 10.1109/TII.2020.3005965.

[58] Connor Shorten and Taghi Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6 (July 2019). DOI: 10.1186/s40537-019-0197-0.

[59] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[60] Steven P Sinkins and Fred Gould. “Gene drive systems for insect disease vectors”. In: *Nature Reviews Genetics* 7.6 (2006), pp. 427–435.

[61] Nitish Srivastava et al. “Dropout: A simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[62] New Zealand Stats New Zealand. *Extinction threat to indigenous land species*. URL: `https://www.stats.govt.nz/indicators/extinction-threat-to-indigenous-land-species` (visited on 06/30/2021).

[63] Christian Szegedy et al. “Going deeper with convolutions”. In: Cvpr. 2015.

[64] Nobuaki Takahashi, Makio Kashino, and Naoyuki Hironaka. “Structure of rat ultrasonic vocalizations and its relevance to behavior”. In: *PloS one* 5.11 (2010), e14115.

[65] Peter M. Vitousek et al. “Human Domination of Earth’s Ecosystems”. In: *Science* 277.5325 (1997), pp. 494–499. ISSN: 0036-8075. DOI: `10.1126/science.277.5325.494`. eprint: `https://science.sciencemag.org/content/277/5325/494.full.pdf`. URL: `https://science.sciencemag.org/content/277/5325/494`.

[66] Sida Wang and Christopher Manning. “Fast dropout training”. In: *international conference on machine learning*. 2013, pp. 118–126.

[67] B Warburton and I Yockney. “Comparison of two luring methods for trapping brushtail possums in non-forest habitats of New Zealand”. In: *New Zealand Journal of Zoology* 36.4 (2009), pp. 401–405.

[68] John Woodrow Winter. “The behaviour and social organisation of the brush-tail possum (Trichosurus vulpecula: Kerr)”. In: (1976).

[69] Jonci N Wolff et al. “Mitonuclear interactions, mtDNA-mediated thermal plasticity and implications for the Trojan Female Technique for pest control”. In: *Scientific reports* 6.1 (2016), pp. 1–7.

[70] Masatoshi Yasuda. “Monitoring diversity and abundance of mammals with camera traps: a case study on Mount Tsukuba, central Japan”. In: *Mammal study* 29.1 (2004), pp. 37–46.

[71] C. Zach, T. Pock, and H. Bischof. "A Duality Based Approach for Realtime TV-L¡sup¿1¡/sup¿ Optical Flow". In: *Proceedings of the 29th DAGM Conference on Pattern Recognition*. Heidelberg, Germany: Springer-Verlag, 2007, pp. 214–223. ISBN: 9783540749332.

[72] Erika S Zavaleta, Richard J Hobbs, and Harold A Mooney. "Viewing invasive species removal in a whole-ecosystem context". In: *Trends in Ecology & Evolution* 16.8 (2001), pp. 454–459.

[73] New Zealand. *Wildlife Act 1953*. URL: `https://www.legislation.govt.nz/act/public/1953/0031/latest/whole.html` (visited on 06/30/2021).

[74] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

Appendix

Exact code layout for high and low quality model implementations

The entire codebase can be found at https://github.com/TheCacophonyProject/classifier-pipeline

```
class ModelCRNN_HQ(ConvModel):
    """
    Convolutional neural network model feeding into an LSTM
    """
    MODEL_NAME = "model_hq"
    MODEL_DESCRIPTION = "CNN + LSTM"
    DEFAULT_PARAMS = {
        # training params
        "batch_size": 16,
        "learning_rate": 1e-4,
        "learning_rate_decay": 1.0,
        "l2_reg": 0.0,
        "label_smoothing": 0.1,
        "keep_prob": 0.5,
        # model params
        "batch_norm": True,
        "lstm_units": 512,
        "enable_flow": True,
        # augmentation
        "augmentation": True,
        "thermal_threshold": 10,
        "scale_frequency": 0.5,
    }

    def model_name(self):
        return ModelCRNN_HQ.MODEL_NAME

    def __init__(self, labels, train_config, training=False,
                 tflite=False, **kwargs):
        """
        Initialise the model
        :param labels: number of labels for model to predict
        """
        super().__init__(train_config=train_config, training=training, tflite=tflite)
        self.params.update(self.DEFAULT_PARAMS)
        self.params.update(kwargs)
        self._build_model(labels)

    def _build_model(self, label_count):
        ########################################
        # CNN + LSTM
        # based on https://arxiv.org/pdf/1507.06527.pdf
        ########################################
        # dimensions are documents as follows
        # B batch size
        # F frames per segment
        # C channels
        # H frame height
        # W frame width
        thermal, flow, mask = self.process_inputs()
        frame_count = tf.shape(self.X)[1]
        # ----------------------------------------
        # run the Convolutions
        layer = thermal
        layer = self.conv_layer("thermal/1", layer, 64, [3, 3], pool_stride=2)
        layer = self.conv_layer("thermal/2", layer, 64, [3, 3], pool_stride=2)
        layer = self.conv_layer("thermal/3", layer, 96, [3, 3], pool_stride=2)
        layer = self.conv_layer("thermal/4", layer, 128, [3, 3], pool_stride=2)
```

```
layer = self.conv_layer("thermal/5", layer, 128, [3, 3], pool_stride=1)
filtered_conv = layer
filtered_out = tf.reshape(
    filtered_conv,
    [-1, frame_count, tools.product(filtered_conv.shape[1:])],
    name="thermal/out",
)
logging.info("Thermal convolution output shape: {}".format(filtered_conv.shape))
if self.params["enable_flow"]:
    # integrate thermal and flow into a 3 channel layer
    layer = tf.concat((thermal, flow), axis=3)
    layer = self.conv_layer("motion/1", layer, 64, [3, 3], pool_stride=2)
    layer = self.conv_layer("motion/2", layer, 64, [3, 3], pool_stride=2)
    layer = self.conv_layer("motion/3", layer, 96, [3, 3], pool_stride=2)
    layer = self.conv_layer("motion/4", layer, 128, [3, 3], pool_stride=2)
    layer = self.conv_layer("motion/5", layer, 128, [3, 3], pool_stride=1)
    motion_conv = layer
    motion_out = tf.reshape(
        motion_conv,
        [-1, frame_count, tools.product(motion_conv.shape[1:])],
        name="motion/out",
    )
    logging.info(
        "Motion convolution output shape: {}".format(motion_conv.shape)
    )

    out = tf.concat((filtered_out, motion_out), axis=2, name="out")
else:
    out = tf.concat((filtered_out,), axis=2, name="out")
logging.info("Output shape {}".format(out.shape))
# ------------------------------------------
# run the LSTM
memory_output, memory_state = self._build_memory(out)
if self.params["l2_reg"] > 0:
    regularizer = tf.keras.regularizers.l2(l=0.5 * (self.params["l2_reg"]))
else:
    regularizer = None
#  to do change to keras dense
# dense = tf.keras.layers.Dense(self.params["lstm_units"])(memory_output)
# # dense hidden layer
dense = tf.compat.v1.layers.dense(
    inputs=memory_output,
    units=self.params["lstm_units"],
    activation=tf.nn.relu,
    name="hidden",
    kernel_regularizer=regularizer,
)
if not self.tflite:
    dense = tf.nn.dropout(dense, rate=1 - (self.keep_prob))
# dense layer on top of convolutional output mapping to class labels.
# logits = tf.keras.layers.Dense(label_count)(dense)
logits = tf.compat.v1.layers.dense(
    inputs=dense,
    units=label_count,
    activation=None,
    name="logits",
    kernel_regularizer=regularizer,
)
tf.compat.v1.summary.histogram("weights/dense", dense)
tf.compat.v1.summary.histogram("weights/logits", logits)
# loss
softmax_loss = tf.compat.v1.losses.softmax_cross_entropy(
    onehot_labels=tf.one_hot(self.y, label_count),
    logits=logits,
    label_smoothing=self.params["label_smoothing"],
    scope="softmax_loss",
)
if self.params["l2_reg"] != 0:
```

```
            reg_loss = tf.compat.v1.losses.get_regularization_loss()
            loss = tf.add(softmax_loss, reg_loss, name="loss")
            tf.compat.v1.summary.scalar("loss/reg", reg_loss)
            tf.compat.v1.summary.scalar("loss/softmax", softmax_loss)
        else:
            # just relabel the loss node
            loss = tf.identity(softmax_loss, name="loss")
        class_out = tf.argmax(input=logits, axis=1, name="class_out")
        correct_prediction = tf.equal(class_out, self.y)
        pred = tf.nn.softmax(logits, name="prediction")
        accuracy = tf.reduce_mean(
            input_tensor=tf.cast(correct_prediction, dtype=tf.float32), name="accuracy"
        )
        # ------------------------------------
        # novelty
        self.setup_novelty(logits, dense)
        self.setup_optimizer(loss)
        # make reference to special nodes
        tf.identity(memory_state, "state_out")
        tf.identity(dense, "hidden_out")
        tf.identity(logits, "logits_out")
        self.attach_nodes()


class ModelCRNN_LQ(ConvModel):
    """
    Convolutional neural network model feeding into an LSTM
    Lower quality, but faster model
    Uses 256 LSTM units and conv stride instead of max pool
    Uses less filters
    Trains on GPU at around 5ms / segment as apposed to 16ms for the high quality model.
    """
    MODEL_NAME = "model_lq"
    MODEL_DESCRIPTION = "CNN + LSTM"
    DEFAULT_PARAMS = {
        # training params
        "batch_size": 16,
        "learning_rate": 1e-4,
        "learning_rate_decay": 1.0,
        "l2_reg": 0,
        "label_smoothing": 0.1,
        "keep_prob": 0.2,
        # model params
        "batch_norm": True,
        "lstm_units": 256,
        "enable_flow": True,
        # augmentation
        "augmentation": True,
        "thermal_threshold": 10,
        "scale_frequency": 0.5,
        "hq": False,
    }
    def model_name(self):
        return ModelCRNN_LQ.MODEL_NAME
    def __init__(self, labels, train_config, training, **kwargs):
        """
        Initialise the model
        :param labels: number of labels for model to predict
        """
        super().__init__(train_config=train_config, training=training)
        self.params.update(self.DEFAULT_PARAMS)
        self.params.update(kwargs)
        if self.params["hq"]:
            self.layers = 5
            self.layer_filters = [64, 64, 96, 128, 128]
            self.conv_stride = [1, 1, 1, 1, 1]
            self.pool_stride = [2, 2, 2, 2, 1]
            self.kernel_size = [3, 3]
```

```
        else:
            # from the pdf this is the layers used
            # self.layers = 3
            # self.layer_filters = [32, 64, 64]
            # self.kernel_size = [[8, 8], [4, 4], [3, 3]]
            # self.pool_stride = [1, 1, 1, 1, 1]
            # self.conv_stride = [4, 2, 1]
            self.layers = 5
            self.layer_filters = [32, 48, 64, 64, 64]
            self.kernel_size = [[3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]
            self.pool_stride = [1, 1, 1, 1, 1]
            self.conv_stride = [2, 2, 2, 2, 1]
        self._build_model(labels)
    def _build_model(self, label_count):
        ######################################
        # CNN + LSTM
        # based on https://arxiv.org/pdf/1507.06527.pdf
        ######################################
        # dimensions are documents as follows
        # B batch size
        # F frames per segment
        # C channels
        # H frame height
        # W frame width
        thermal, flow, mask = self.process_inputs()
        frame_count = tf.shape(self.X)[1]
        # ---------------------------------------
        # run the Convolutions
        layer = thermal
        for i in range(self.layers):
            layer = self.conv_layer(
                "thermal/{}".format(i),
                layer,
                self.layer_filters[i],
                self.kernel_size[i],
                conv_stride=self.conv_stride[i],
                pool_stride=self.pool_stride[i],
            )
        filtered_conv = layer
        logging.info("Thermal convolution output shape: {}".format(filtered_conv.shape))
        filtered_out = tf.reshape(
            filtered_conv,
            [-1, frame_count, tools.product(filtered_conv.shape[1:])],
            name="thermal/out",
        )
        if self.params["enable_flow"]:
            # integrate thermal and flow into a 3 channel layer

            layer = tf.concat((thermal, flow), axis=3)

            for i in range(self.layers):
                layer = self.conv_layer(
                    "motion/{}".format(i),
                    layer,
                    self.layer_filters[i],
                    self.kernel_size[i],
                    conv_stride=self.conv_stride[i],
                    pool_stride=self.pool_stride[i],
                )
            motion_conv = layer
            logging.info(
                "Motion convolution output shape: {}".format(motion_conv.shape)
            )
            motion_out = tf.reshape(
                motion_conv,
                [-1, frame_count, tools.product(motion_conv.shape[1:])],
                name="motion/out",
            )
```

```
            out = tf.concat((filtered_out, motion_out), axis=2, name="out")
        else:
            out = tf.concat((filtered_out,), axis=2, name="out")
        logging.info("Output shape {}".format(out.shape))
        memory_output, memory_state = self._build_memory(out)
        logging.info("memory_output output shape: {}".format(memory_output.shape))
        logging.info("memory_state output shape: {}".format(memory_state.shape))
        # -----------------------------------
        # dense / logits
        # dense layer on top of convolutional output mapping to class labels.
        logits = tf.keras.layers.Dense(label_count)(memory_output)
        tf.compat.v1.summary.histogram("weights/logits", logits)
        softmax_loss = tf.compat.v1.losses.softmax_cross_entropy(
            onehot_labels=tf.one_hot(self.y, label_count),
            logits=logits,
            label_smoothing=self.params["label_smoothing"],
            scope="softmax_loss",
        )
        if self.params["l2_reg"] != 0:
            with tf.compat.v1.variable_scope("logits", reuse=True):
                logit_weights = tf.compat.v1.get_variable("kernel")
            reg_loss = (
                tf.nn.l2_loss(logit_weights, name="loss/reg") * self.params["l2_reg"]
            )
            loss = tf.add(softmax_loss, reg_loss, name="loss")
            tf.compat.v1.summary.scalar("loss/reg", reg_loss)
            tf.compat.v1.summary.scalar("loss/softmax", softmax_loss)
        else:
            # just relabel the loss node
            loss = tf.identity(softmax_loss, name="loss")
        class_out = tf.argmax(input=logits, axis=1, name="class_out")
        correct_prediction = tf.equal(class_out, self.y)
        pred = tf.nn.softmax(logits, name="prediction")
        accuracy = tf.reduce_mean(
            input_tensor=tf.cast(correct_prediction, dtype=tf.float32), name="accuracy"
        )
        self.setup_novelty(logits, memory_output)
        self.setup_optimizer(loss)
        # make reference to special nodes
        tf.identity(memory_state, "state_out")
        tf.identity(memory_output, "hidden_out")
        tf.identity(logits, "logits_out")
        self.attach_nodes()

class Model_CNN(ConvModel):
    """
    Convolutional neural network model feeding into an LSTM
    Trains on GPU at around 5ms / segment as apposed to 16ms for the high quality model.
    """
    MODEL_NAME = "model_cnn"
    MODEL_DESCRIPTION = "CNN"
    DEFAULT_PARAMS = {
        # training params
        "batch_size": 16,
        "learning_rate": 1e-4,
        "learning_rate_decay": 1.0,
        "l2_reg": 0,
        "label_smoothing": 0.1,
        "keep_prob": 0.2,
        # model params
        "batch_norm": True,
        "enable_flow": True,
        # augmentation
        "augmentation": True,
        "thermal_threshold": 10,
        "scale_frequency": 0.5,
        "hq": False,
    }
```

```
def model_name(self):
    return Model_CNN.MODEL_NAME
def __init__(self, labels, train_config, training, tflite, **kwargs):
    """
    Initialise the model
    :param labels: number of labels for model to predict
    """
    super().__init__(train_config=train_config, training=training, tflite=tflite)
    self.frame_count = 1
    # number of frames per segment during training
    self.training_segment_frames = 1
    # number of frames per segment during testing
    self.testing_segment_frames = 1

    self.params.update(self.DEFAULT_PARAMS)
    self.params.update(kwargs)
    if self.params["hq"]:
        self.layers = 5
        self.layer_filters = [64, 64, 96, 128, 128]
        self.conv_stride = [1, 1, 1, 1, 1]
        self.pool_stride = [2, 2, 2, 2, 1]
        self.kernel_size = [3, 3]
    else:
        self.layers = 5
        self.layer_filters = [32.48, 64, 64, 64]
        self.kernel_size = [3, 3]
        self.pool_stride = [1, 1, 1, 1, 1]
        self.conv_stride = [2, 2, 2, 2, 1]
    self._build_model(labels)
def _build_model(self, label_count):
    ######################################
    # CNN + LSTM
    # based on https://arxiv.org/pdf/1507.06527.pdf
    ######################################

    # dimensions are documents as follows
    # B batch size
    # F frames per segment
    # C channels
    # H frame height
    # W frame width

    thermal, flow, mask = self.process_inputs()
    # ----------------------------------
    # run the Convolutions

    layer = thermal

    for i in range(self.layers):
        layer = self.conv_layer(
            "thermal/{}".format(i),
            layer,
            self.layer_filters[i],
            self.kernel_size,
            conv_stride=self.conv_stride[i],
            pool_stride=self.pool_stride[i],
        )

    filtered_conv = layer
    logging.info("Thermal convolution output shape: {}".format(filtered_conv.shape))

    if self.params["enable_flow"]:
        # integrate thermal and flow into a 3 channel layer

        layer = tf.concat((thermal, flow), axis=3)

        for i in range(self.layers):
            layer = self.conv_layer(
```

```
                "motion/{}".format(i),
                layer,
                self.layer_filters[i],
                self.kernel_size,
                conv_stride=self.conv_stride[i],
                pool_stride=self.pool_stride[i],
            )

        motion_conv = layer
        logging.info(
            "Motion convolution output shape: {}".format(motion_conv.shape)
        )
        motion_out = tf.reshape(
            motion_conv,
            [-1, self.frame_count, tools.product(motion_conv.shape[1:])],
            name="motion/out",
        )
        filtered_out = tf.reshape(
            filtered_conv,
            [-1, self.frame_count, tools.product(filtered_conv.shape[1:])],
            name="thermal/out",
        )
        out = tf.concat((filtered_out, motion_out), axis=2, name="out")
    else:
        out = tf.compat.v1.layers.flatten(filtered_conv)
    logging.info("Output shape {}".format(out.shape))

    # dense / logits

    # dense layer on top of convolutional output mapping to class labels.
    logits = tf.keras.layers.Dense(label_count)(out)
    print("logits.shape", logits.shape)
    tf.compat.v1.summary.histogram("weights/logits", logits)
    softmax_loss = tf.compat.v1.losses.softmax_cross_entropy(
        onehot_labels=tf.one_hot(self.y, label_count),
        logits=logits,
        label_smoothing=self.params["label_smoothing"],
        scope="softmax_loss",
    )
    if self.params["l2_reg"] != 0:
        with tf.compat.v1.variable_scope("logits", reuse=True):
            logit_weights = tf.compat.v1.get_variable("kernel")

        reg_loss = (
            tf.nn.l2_loss(logit_weights, name="loss/reg") * self.params["l2_reg"]
        )
        loss = tf.add(softmax_loss, reg_loss, name="loss")
        tf.compat.v1.summary.scalar("loss/reg", reg_loss)
        tf.compat.v1.summary.scalar("loss/softmax", softmax_loss)
    else:
        # just relabel the loss node
        loss = tf.identity(softmax_loss, name="loss")
    class_out = tf.argmax(input=logits, axis=1, name="class_out")
    correct_prediction = tf.equal(class_out, self.y)
    pred = tf.nn.softmax(logits, name="prediction")
    accuracy = tf.reduce_mean(
        input_tensor=tf.cast(correct_prediction, dtype=tf.float32), name="accuracy"
    )

    self.setup_novelty(logits, out)
    self.setup_optimizer(loss)

    # make reference to special nodes
    # not used for anything as we aren't doing RNN for tflite
    tf.identity(out, "hidden_out")
    tf.identity(logits, "logits_out")
    self.attach_nodes()
```